NPS-ME-96-27

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

SIMULATION OF SMALL ROBOTIC VEHICLE PERFORMANCE DURING UXO GATHERING OPERATIONS USING DISCRETE EVENT CONTROL

by

Todd A. Lewis

September, 1996

Thesis Advisor: Anthoney J. Healey

Approved for public release: distribution is unlimited.

Prepared for:
NAVEODTECHDIV R&D Division
Indian Head, Md 20640

Naval Postgraduate School
Monterey, California 93943

Rear Admiral Marsha Johnson Evans
Superintendent

This thesis was prepared in conjunction with research sponsored in part by NAVEODTECDIV R&D Dept., Indian Head, MD under N0464A96WR00778.

Released by:

David W. Netzer, Associate Provost
And Dean of Research

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE **SIMULATION OF SMALL ROBOTIC VEHICLE PERFORMANCE DURING UXO GATHERING OPERATIONS USING DISCRETE EVENT CONTROL**

5. FUNDING NUMBERS N0464A96WR00778

6. AUTHOR Todd A. Lewis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION NPS-ME-96-27

9. SPONSORING/MONITORING AGENCY NAME(S) AND aDDRESS(ES)
NAVEODTECDIV R&D Dept, Indian,Head , Md 20640

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT

The subject of minefield clearance has received much attention in the last few years due to it's necessity during modern warfighting, and it's unavoidable inherent dangers. Clearing the battlefield of Unexploded Ordnance (UXO) is a formidable task that presently requires the risk human life. Future strategy calls for the use of a fleet of small, inexpensive, but very capable robots to clear the battlefield of all Unexploded Ordnance. The Navy's Explosive Ordnance Disposal Technical Center has developed a "BUGS" Basic UXO Gathering System, in order to examine such strategy. In support of this effort, simulations are being conducted to examine the effects of navigation, control schemes, and terrain characteristics on battlefield clearance operations.

14. SUBJECT TERMS BUGS, Unexploded Ordnance Gathering, Simulation, Hybrid Control System	15. NUMBER OF PAGES 74
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

SIMULATION OF SMALL ROBOTIC VEHICLE PERFORMANCE DURING UXO GATHERING OPERATIONS USING DISCRETE EVENT CONTROL

Todd A. Lewis
Lieutenant, United States Navy
B.S., Norfolk State University, 1989

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September, 1996

Author: Todd A Lewis
Todd A. Lewis

Approved by: Anthony J. Healey
Anthony J. Healey, Thesis Advisor

T R McNelly
Terry McNelly, Chairman
Department of Mechanical Engineering

ABSTRACT

The subject of minefield clearance has received much attention in the last few years due to it's necessity during modern warfighting, and it's unavoidable inherent dangers. Clearing the battlefield of Unexploded Ordnance (UXO) is a formidable task that presently requires the risk of human life. Future strategy calls for the use of a fleet of small, inexpensive, but very capable robots to clear the battlefield of all Unexploded Ordnance. The Navy's Explosive Ordnance Disposal Technical Center has developed a "BUGS" Basic UXO Gathering System in order to examine such strategy. In support of this effort, simulations are being conducted to examine the effects of navigation, control schemes, and terrain characteristics on battlefield clearance operations.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to ackknoledge the financial suport of the NAVEODTECHDIV for funding this thesis. I would also like to thank Draper Laboratory, Foster Miller, and IS Robotics for providing all pictures and technical information.

I want to express my everlasting gratitude to Professor Healey for all his assistance on this thesis. His patience and expert knowledge made my thesis experience a pleasant one. Picking the thesis advisor and not the thesis topic, was some of the best advice ever told to me. It certainly paid off this time. It was an absolute pleasure personally and professionally to work with you.

Lastly, and most importantly, I would like to thank my family for their support and understanding why I was always gone.

I. INTRODUCTION

A. THE PROBLEM

The problem of minefield clearance following dispersion by some type of air delivery system is a challenging task. Hundreds of submunitions can be randomly distributed over a sizeable area in seconds by one air weapon. The Navy's Tomahawk Missile System can deliver hundreds of bomblet submunitions over an area the size of a football field in a matter of seconds. Assuming a small percentage dud rate, this could still leave tens of unexploded submunitions from one missile. Naturally, from multiple Tomahawk missions there could be thousands of unexploded summations left on the battlefield.

Current battlefield clearance techniques call for manned squads to identify, carry away, or blow in place unexploded ordnance (UXO). Because of the inherent danger of these operations, serious injury and loss of life are sometimes the product of battlefield clearance. For example, during the Gulf War both the Army and Marines lost personnel due to submunition clearance operations. This alone indicates the desirability of an autonomous submunition clearance system that requires no human presence on the field during clearing operations.

B. THE CHALLENGE, AND CURRENT ISSUES

The underlying challenge is to safely and effectively clear the battlefield of unexploded ordnance using small and inexpensive robots, reducing human casualties. This challenge has recently been addressed by The Navy's Explosive Ordnance Disposal (EOD)

unexploded ordnance using small and inexpensive robots, reducing human casualties. This challenge has recently been addressed by The Navy's Explosive Ordnance Disposal (EOD) Community. As a result, it is believed that small robotic systems will be used in future battlefield clearance operations. However, this problem is not one that is easily solved. The use of robots requires sophisticated sensors for munition detection and obstacle avoidance, not to mention some type of precise navigation system. Once the UXO has been detected it must then be approached, picked up, neutralized, and then safely removed to a disposal location. In addition, terrain characteristics need to be considered. Terrain characteristics play a very important role in clearance operations. Wheel slip due to muddy, uneven, and rough terrain can greatly reduce clearance time and pose problems for control algorithms. It is very apparent that these operations are very dangerous, and a sophisticated robotic system can easily be lost or severely damaged. If the cost of these robotic systems can be minimized to a level that would be cost effective to mass produce while taking into account the subsequent loss of assets during clearing operations, this problem would be well on its way to being solved. Work is in progress at NAVEODTECHDIV to develop a system of radio controlled teloperated vehicles known as RECORM, to provide sophisticated reconnaissance in EOD clearance scenarios. This platform would have the ability to provide video images of the battlefield to a remote operator who could then identify UXOs for clearance. It is believed a less expensive and capable version, denoted "BUGS" (Basic Unexploded Ordnance Gathering System), should be developed to do actual ordnance handling, or be equipped with Blow In Place (BIP) counter charges. The BUGS platform should be inexpensive enough to account for the occasional loss of assets due to munition

mine field and UXO clearance environments to establish their effectiveness and primary operating parameters. This work examines the performance of such robotic systems by simulating UXO gathering operations using SIMULINK as the primary modeling tool.

C. THESIS SCOPE

The purpose of this thesis is as follows:

1. Build a real-time computer model of a "BUGS" vehicle that accurately simulates dynamic and kinematic performance in UXO clearance operations (using SIMULINK).

2. Develop a mission / vehicle controller (using petri net methodology to model the mission control) and develop its interfaces within the SIMULINK environment.

3. Examine higher level hybrid control scheme effectiveness in modeling robotic vehicle performance.

4. Examine "BUGS" vehicle performance in ideal conditions (perfect navigation/no wheel slip) and with slip induced by varying terrain characteristics .

If a vehicle is not equipped with a reasonably precise navigation system and has a high degree of wheel slip over rough terrain, it is expected that there will be a substantial degradation in mission effectiveness. The key concern of this work is to examine the effects of wheel slip and non-perfect navigation on clearance operations.

II. MODERN AUTONOMOUS ROBOTIC LAND VEHICLE TECHNOLOGY

A. HISTORY

There has been study and experimentation in the area of autonomous robotic vehicles for the past few decades. With expanding technology in electronic circuitry and devices, autonomous robots are becoming more alluring to government and industry as a way to accomplish dangerous and tedious tasks. Robotic vehicles have been designed in many shapes and forms, depending on their intended application. Mobile tracked vehicles have been popular for military use as tanks and personnel carriers. Tracks are effective on the battlefield because they can travel over rough terrain and have a near zero turning radius. Conversely, walking machines, going back to the 1960's, have the advantage of a small footprint, decreasing the chance of inadvertently detonating ordnance during clearance operations. Even though walking vehicles offer many advantages, integrating computer hardware and software with the additional complex mechanical structure is very costly. Conceptually, it is easier to integrate computers and machinery on wheeled and tracked vehicles.

B. CURRENT TECHNOLOGY

Numerous vehicle concepts are being researched by groups such as ARPA, NASA, and the United States Army and Navy. They are categorized by their propulsion means (i.e., wheeled ,walking, or tracked). They are also categorized by there intended environmental use (i.e., land based or surf zone).

1. Tracked Vehicles

One surf zone tracked vehicle currently under development is the "Lemming", manufactured by Foster Miller Inc., shown in Figure 2.1. This is an example of a relatively inexpensive autonomous vehicle for mine countermeasures that carries an internal explosive charge, and is capable of random search operations. It uses onboard magnetic sensors to detect the mine or the UXO. Once UXO detection has occurred, an explosive charge is placed next to the target to be detonated at a later time. In the minefield scenario, the lemming would park itself next to the mine and wait for the self detonation signal. The "Lemming" has two tactile sensors mounted on the left and right sides (front) to detect objects or potential obstacles. Vibration signatures received by the tactile sensor determine the material characteristic of the object (ie: metal, rock, or plastic) [Ref.1].

A second tracked vehicle currently under development is the "Fetch" vehicle, designed by IS Robotics, shown in Figure 2.2. This autonomous robot is equipped with IR sensors to detect obstacles, and magnetic sensors underneath to detect munitions. Once the munition is detected, the carrier arm (with attached magnet) is lowered to pick it up. The munition is then taken the disposal location. The "Fetch" vehicle uses a GPS Navigation System for 0.2 centimeter navigation accuracy.

2. Walking Vehicles

An example of a walking machine is the "Hermes Robot" from IS Robotics. This vehicle is shown in Figure 2.3. Hermes is a compact, walking research robot intended for exploration of small environments. Its legged mobility design allows it to sense the terrain as well as detect obstacles in itr path. Sensors included in this robot are: proximity IR for

navigation, a pitch and roll inclinometer, and whiskers to detect oncoming obstacles.[Ref 2]

3. Wheeled Vehicles

Wheeled robots currently under development include the "RECORM" and the "Micro Rover" vehicles, the latter of which is shown in Figure 2.4. The "RECORM" (Remote Controlled Reconnaissance Monitor) by Navy EODTECHDIV, is designed to provide remote monitoring capability or site survey of hazardous environments. It can be controlled by fiber optics or RF link. The "Micro Rover" by Draper Laboratories, is designed for high speed munition clearance. It is outfitted with sonar and laser sensors for detection. It also features a custom robotic arm for munition handling.

The Swiss Federal Institute of Technology has developed an autonomous robotic land vehicle called the " PREMEX-BE" (PErsonal Mine Explorer"). This vehicle, shown in Figure 2.5, is equipped with a sensor on an extended arm which is connected to a wheeled control package. Direct current motors operate large wheels in an alternating sequence, producing a sweeping motion of the sensor.

The future of innovative systems for mine countermeasures on land lies in the coordinated use of multiples of small robots - each being a very low cost item - but controlled and organized in such a way that human intervention with the deadly item is not direct.

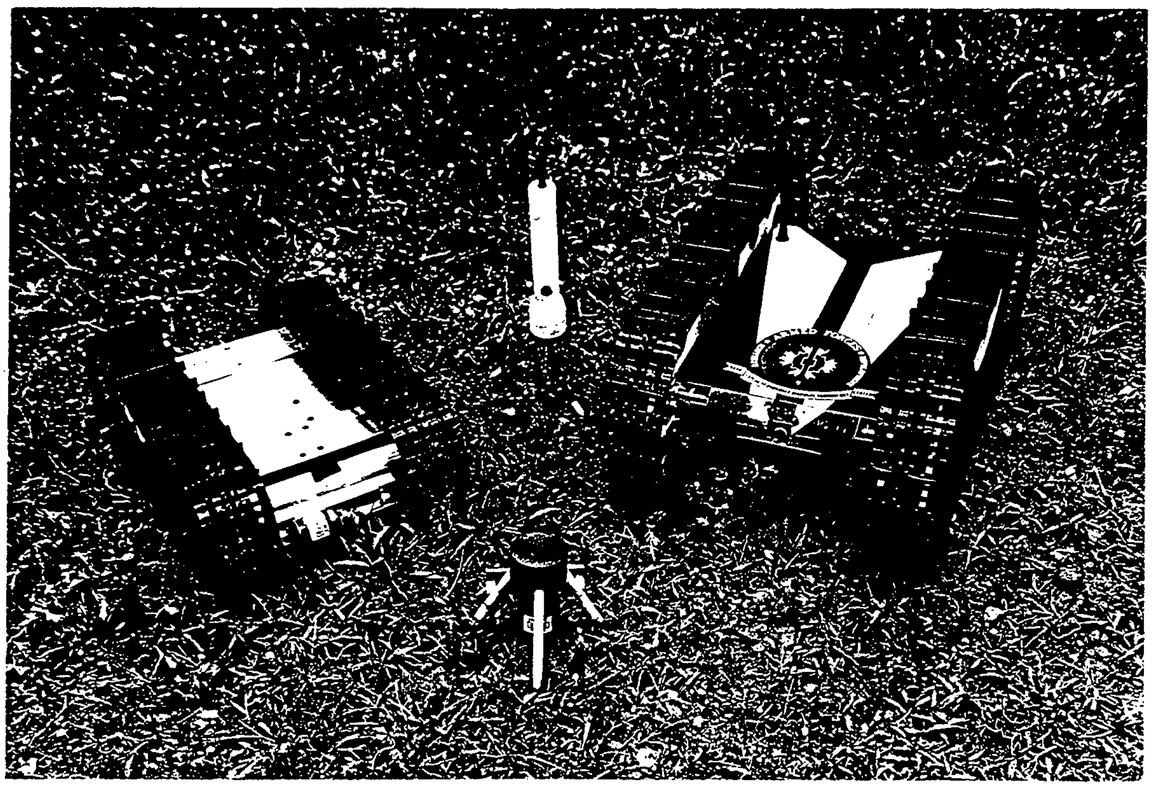


Figure 2.1 Foster Miller Inc "LEMMING" Vehicle

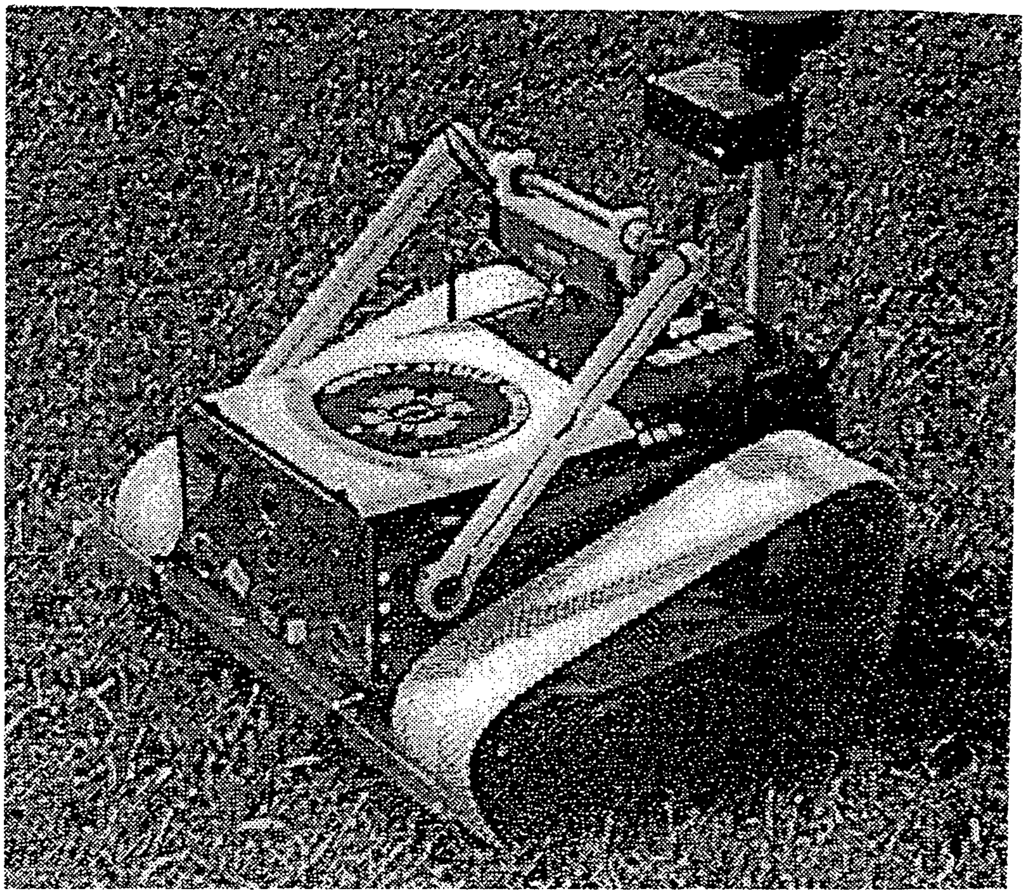


Figure 2.2 IS Robtics "Pebbles" Vehicle

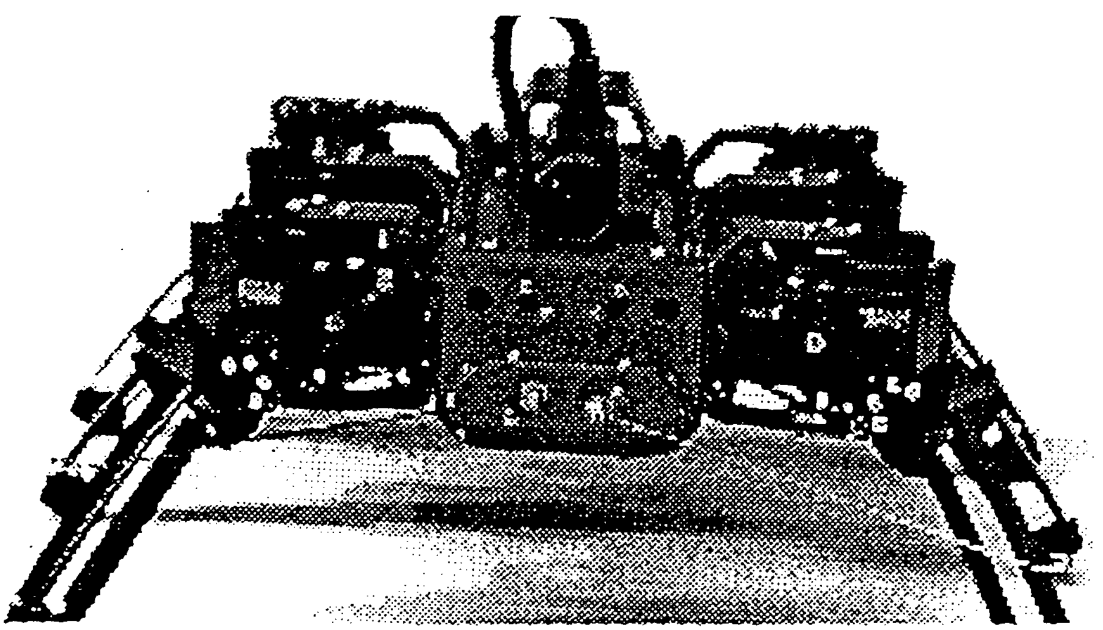


Figure 2.3 IS Robotics "Hermes" Robot

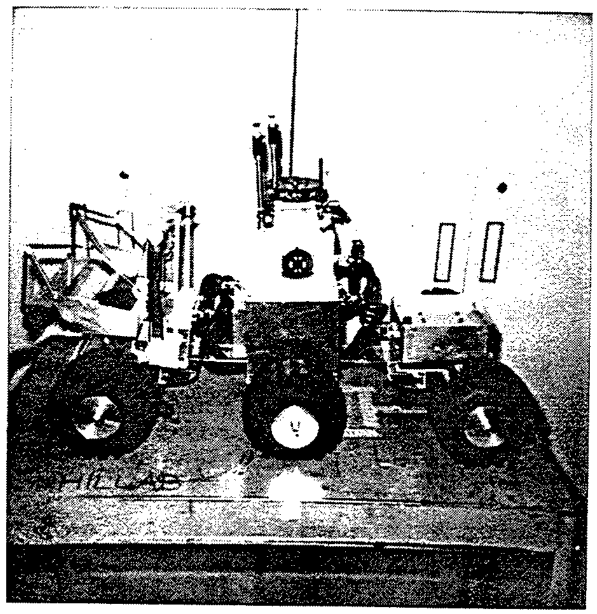


Figure 2.4 Draper Laboratories "MICRO-ROVER"

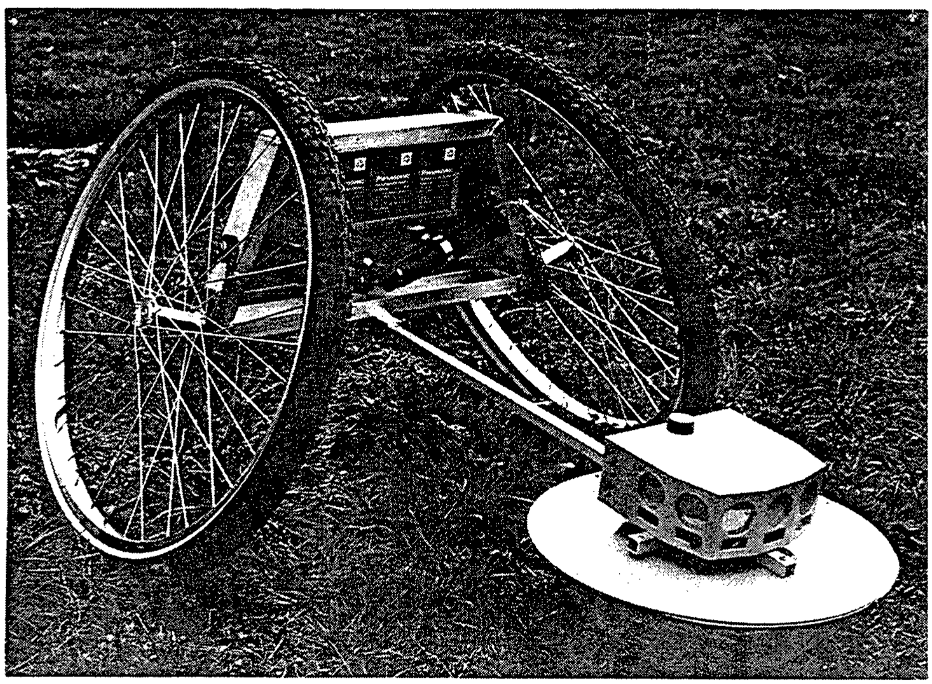


Figure 2.5 SWISS Federal Institute of Technology "PEMEX-BE" Vehicle

III. VEHICLE MODEL DESIGN USING MATLAB/SIMULINK

A. MODELING TOOLS

1. MATLAB

Linear control system analysis and design begins with mathematical models of real systems. These models are representations of such items as machinery, electrical circuits, and aircraft, used to study the dynamic response of real systems. MATLAB (MATrix LABoratory) is a mathematically oriented computer program used for science and engineering applications. MATLAB uses models in the form of continous state differential or difference equations that represent the evolving changes in state as a response to input excitation [Ref 3]. Linear systems can be expressed by frequecy domain transfer functions or time domain state-space equations, allowing "classical" and "modern" control system analysis and design analysis techniques. Either model form can be expressed in continuous (analog) or discrete-time (digital) form. Since the use of digital microprocessors are now common control elements, it is the later that has found significant recent importance. Non-linear systems where the time derivatives of stste are nonlinear functions of the state itself, can be simulated using numerical integration of the nonlinear differential equations of the model. Linear state-space system models are particularly well suited to MATAB since they are matrix based, although nonlinear simulations are also easily conducted in the Matlab environment using the graphically based SIMULINK features.

2. SIMULINK

SIMULINK is a MATLAB graphically based software package for modeling, simulating, and analyzing dynamical systems. It supports linear and nonlinear systems modeled in continuous time, sampled time, or a hybrid of the two. The SIMULINK environment provides a graphical user interface (GUI) for building models in the form of block diagrams that are easily modified and saved. It is equipped with a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. After the model is defined, a simulation can be executed using a number of integration methods.

B. CONTROL SYSTEMS

1. Discrete-Time Control Systems

In recent years, because of the expended capabilities of the microprocessor, there has been an increase in the use of digital controllers for control system implementation. The application of digital control has made possible "intelligent" motion in industrial robots, the optimization of fuel economy in automobiles, and refinements in the operation of household appliances such as microwave ovens. Decision-making capability and flexibility in control programming are major advantages of digital control systems. Discrete-time control systems are control systems in which one or more continuous time variables can be sampled / or changed at discrete instants of time. These instants are specified times at which some measurement is taken. The time interval

between consecutive discrete intervals is taken to be very short so that the data between the time interval can be approximated by simple interpolation algorithms. Signals from discrete-time control systems are in sampled-data (digital) form. Figure 3.1 shows a discrete time signal. If a digital computer is involved in a control system as a digital controller, sampled data must be converted into a finite precision digital data format (usually 12 / 14 / 16 bit binary format). The discrete-time controller samples the continuous-time signal at the discrete time points. The term "discretization" rather than "sampling" is frequently used, although both mean basically the same thing. The stability requirements and design of discrete time digital control algorithms are well known (Ogata, 1996), and it is also well known that the control parameters depend on the update rate (rate at which sensory data are sampled).

2. Continuous-Time / Discrete Time / Discrete State Control Systems

Continuous-time systems may be described by differential equations as opposed to the use of difference equations with discrete-time systems. Continuous-time systems compute the value of the derivative of the state vector at the current time which is always assumed to be a continuous function of itme, rather than the value of the state vector at the next sample time, which discrete-time state space equations compute. A continuous-time signal is defined over a continuous range of time. The amplitude may assume a continuous range of values or a finite set of values. Figure 3.2 shows an example of a continuous time signal. Some applications of continuous-time controllers include, motor speed governors, cruise control systems, and home heating systems. A hybrid system is one where some states are continuous while othere such as the set of

mission states (for example, search mode, transit mode, pick up, etc.) are discrete, taking essentially binary values to determine if any one or another is active or not.

C. VEHICLE MODEL

1. Components

The SIMULINK "BUGS" model is a "hybrid", state-space, discrete event controller for a single vehicle. Given target location and commanded speed, it uses a line of sight guidance control scheme to transitthe vehicle to the goal point. A full block diagram model can be seen in Appendix A, along with a vehicle profile/equation sheet. The model is sub-divided into three component areas:

a. Maneuvering and Drive Component

The maneuvering and drive component shown in Figure 3.3, takes commanded vehicle velocity and commanded turn rate input from the controller and computes commanded wheel speed. This commanded wheel speed is then input into a difference equation represented in SIMULINK by a discrete-time (z transform) block diagram, where the updated wheel speed is computed. Individual track speed and overall vehicle speed is then computed from the wheel speed input. The following equations apply;

Commanded Wheel Speed # 1: $w1c(k) = ucom(k)/d + rcom(k)*D/d$

Commanded Wheel Speed # 2: $w2c(k) = ucom(k)/d - rcom(k)*D/d$

Wheel Speed # 1 $w_1(k+1) = a*w1(k) + (1-a)*w1c(k)$

Wheel Speed # 2: $w2(k+1) = a*w2(k) + (1-a)*w2c(k)$

Track Speed #1: $v1(k) = w1(k)*D/d$

Track Speed #2: $v2(k) = w2(k)*D/d$

Vehicle Speed: $u(k) = 0.5*(v1(k) - v2(k))$

where *ucom* = commanded velocity (meters/sec)

rcom = commanded turn rate (rad/sec)

D = vehicle diameter (meters)

d = wheel diameter (meters)

a = motor lag constant

b. Heading and Position (Navigation) Component

The heading component shown in Figure 3.4, takes inputs of turn rate and current heading and computes an updated heading. This component also uses a difference equation, represented in SIMULINK by a discrete-time (z transform)block diagram for computation. The position component shown in Figure 3-4, takes inputs of current position, vehicle speed, and heading to compute updated X & Y positions. Again, this also uses a difference equation and is represented in SIMULINK by a discrete-time (z transform) block diagram. The following equations are used for this component;

Heading: $\Psi(k+1) = \Psi(k) + dt*r(k)$

X coordinate: $X(k+1) = X(k) + u(k)*cos(\Psi(k))$

Y coordinate: $Y(k+1) = Y(k) + u(k)*sin(\Psi(k))$

where $r(k)$ = current turn rate (rad/sec)

c. Controller Component

The controller shown in Figure 3.5, is the central processing element of the "BUGS" model. It is designed as finite state automaton. The controller takes 21 data inputs and multiplexes them into a vector which is input into a MATLAB file representing the mission control logic for the vehicle. The MATLAB file reads in position, heading, time, dynamic states, and transition states, then computes updated values based on a series of 'transition' statements. These updated values are put into a vector and de-multiplexed back into the system. The state logic diagram shown in Figure 3.6, represents the functional behavior which the model follows to determine the current mission state, and give commands for the next.

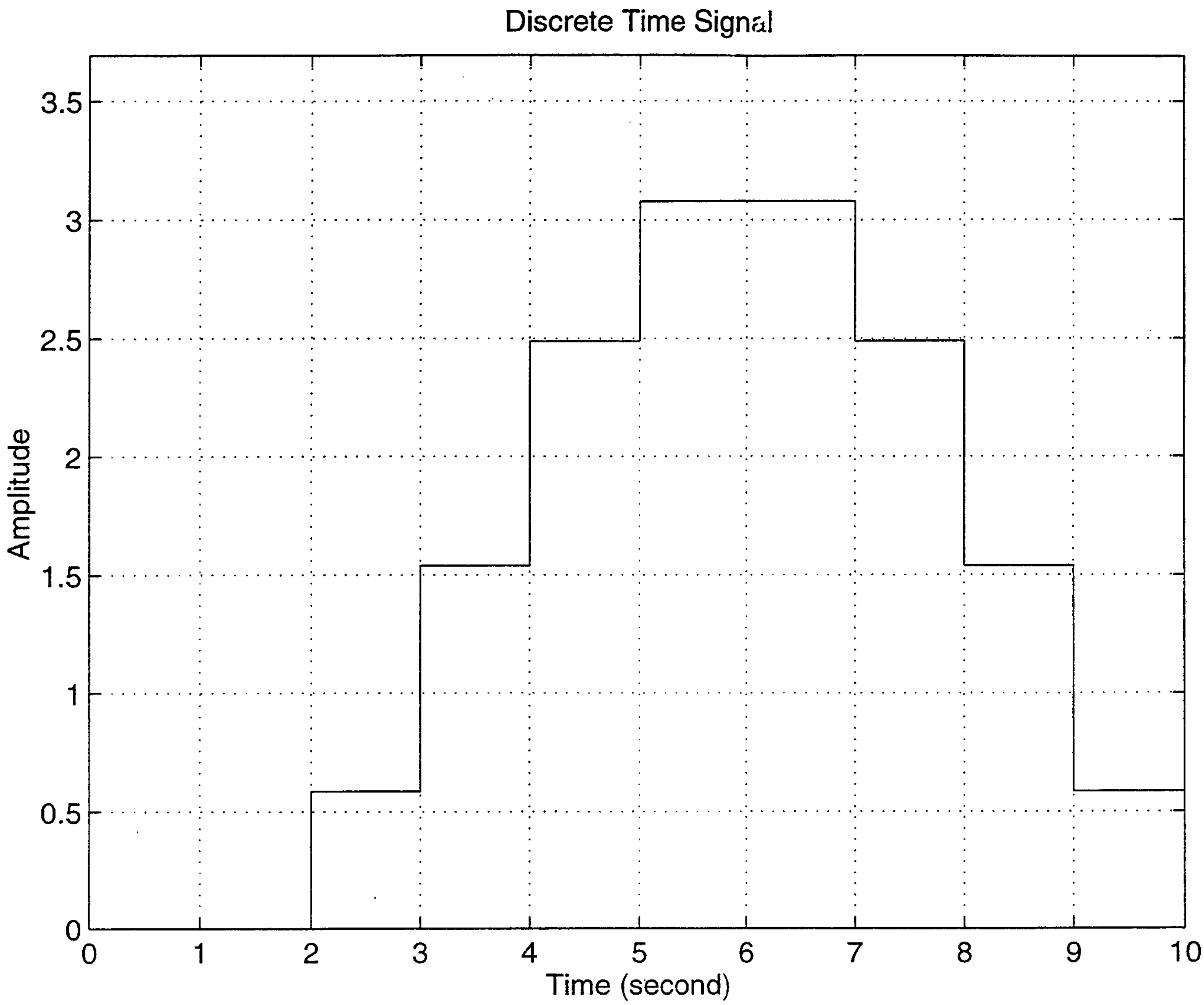


Figure 3.1 Discrete Time Signal

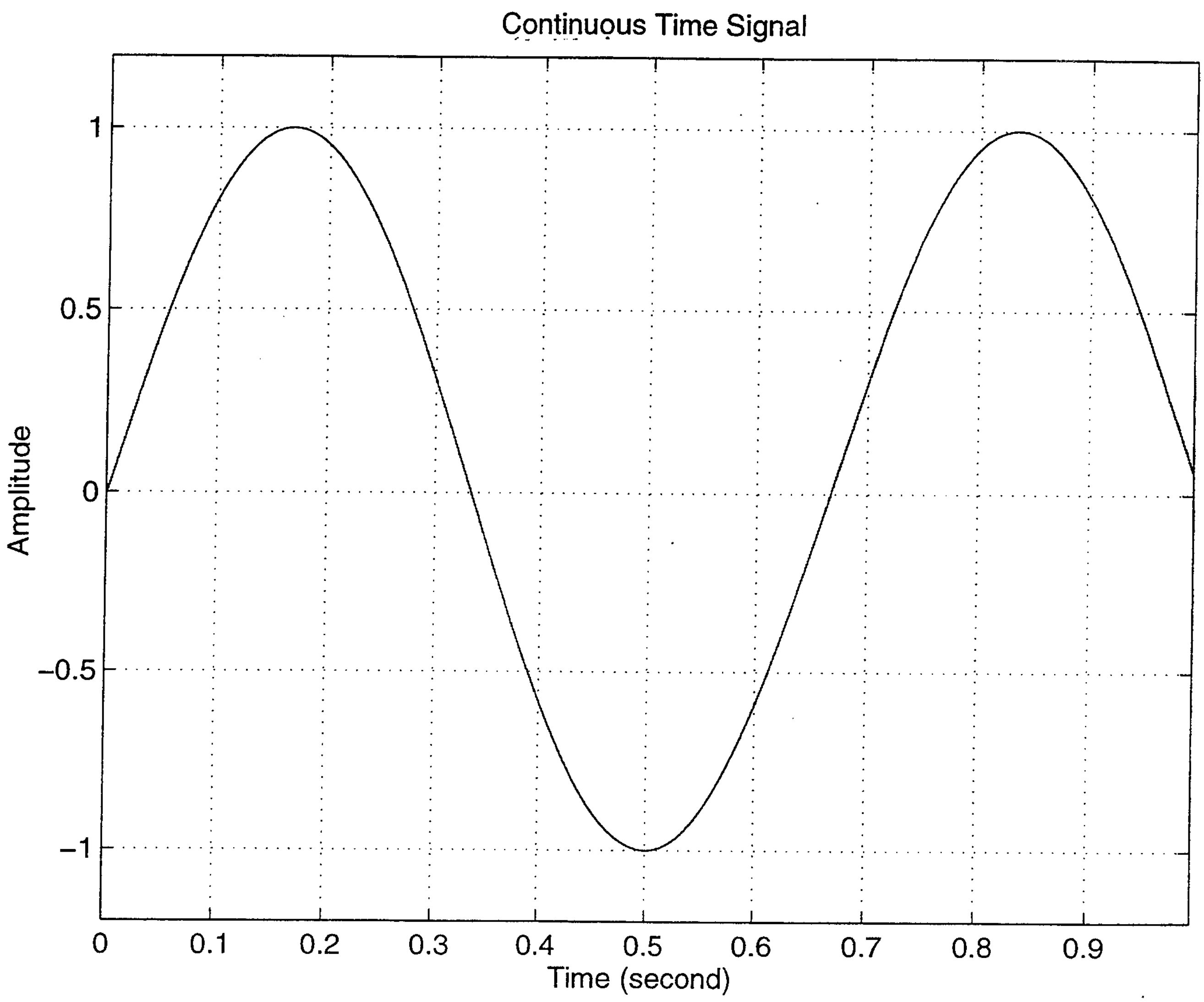


Figure 3.2 Continuous Time Signal

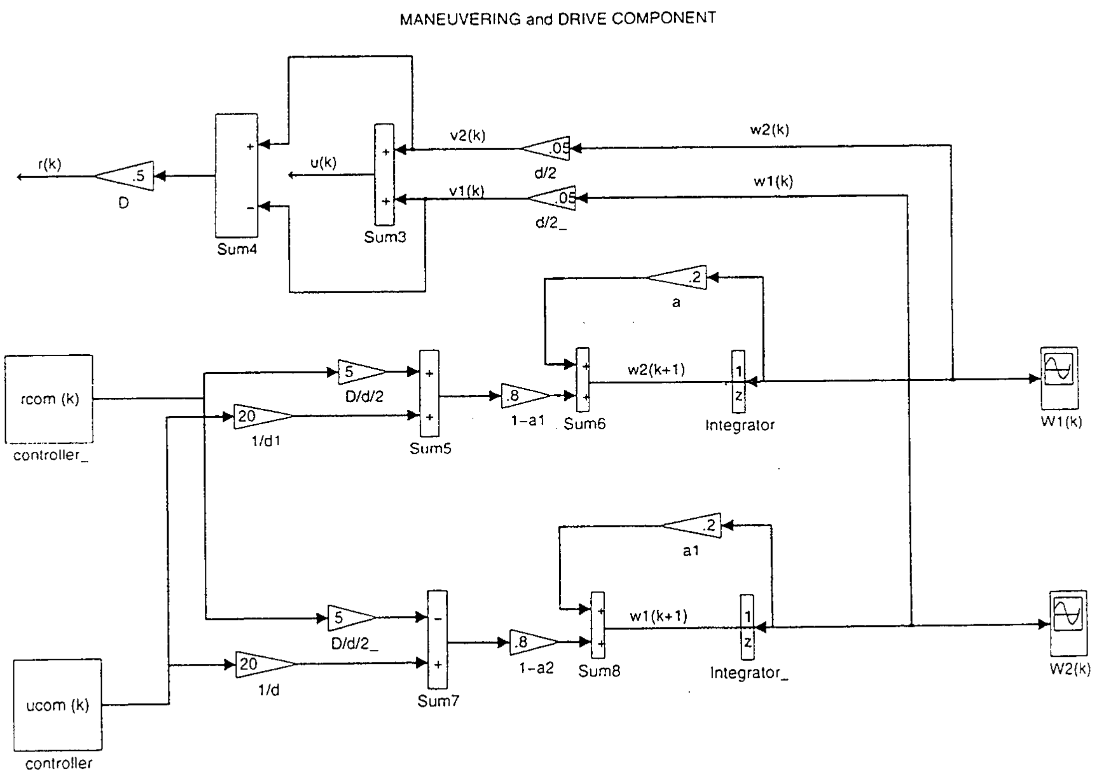


Figure 3.3 Block Diagram "Maneuvering and Drive Component"

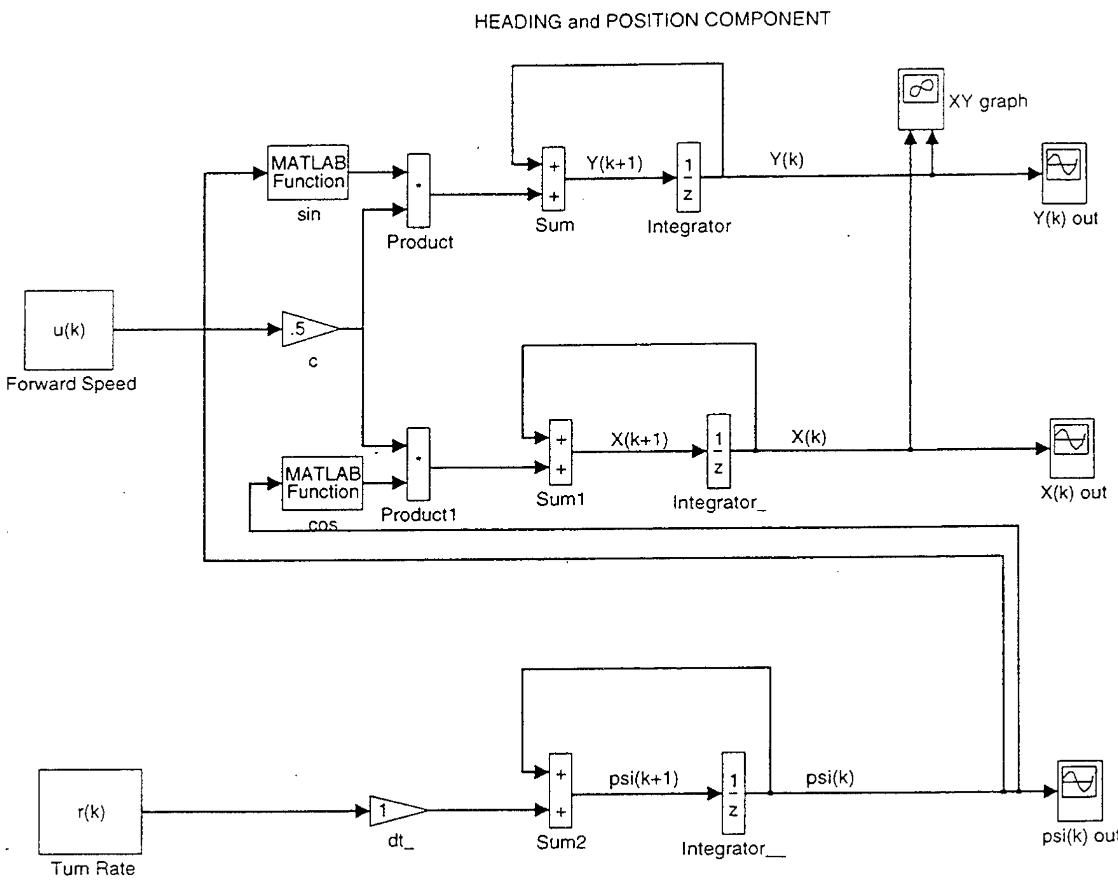


Figure 3.4 Block Diagram " Heading and Position Component"

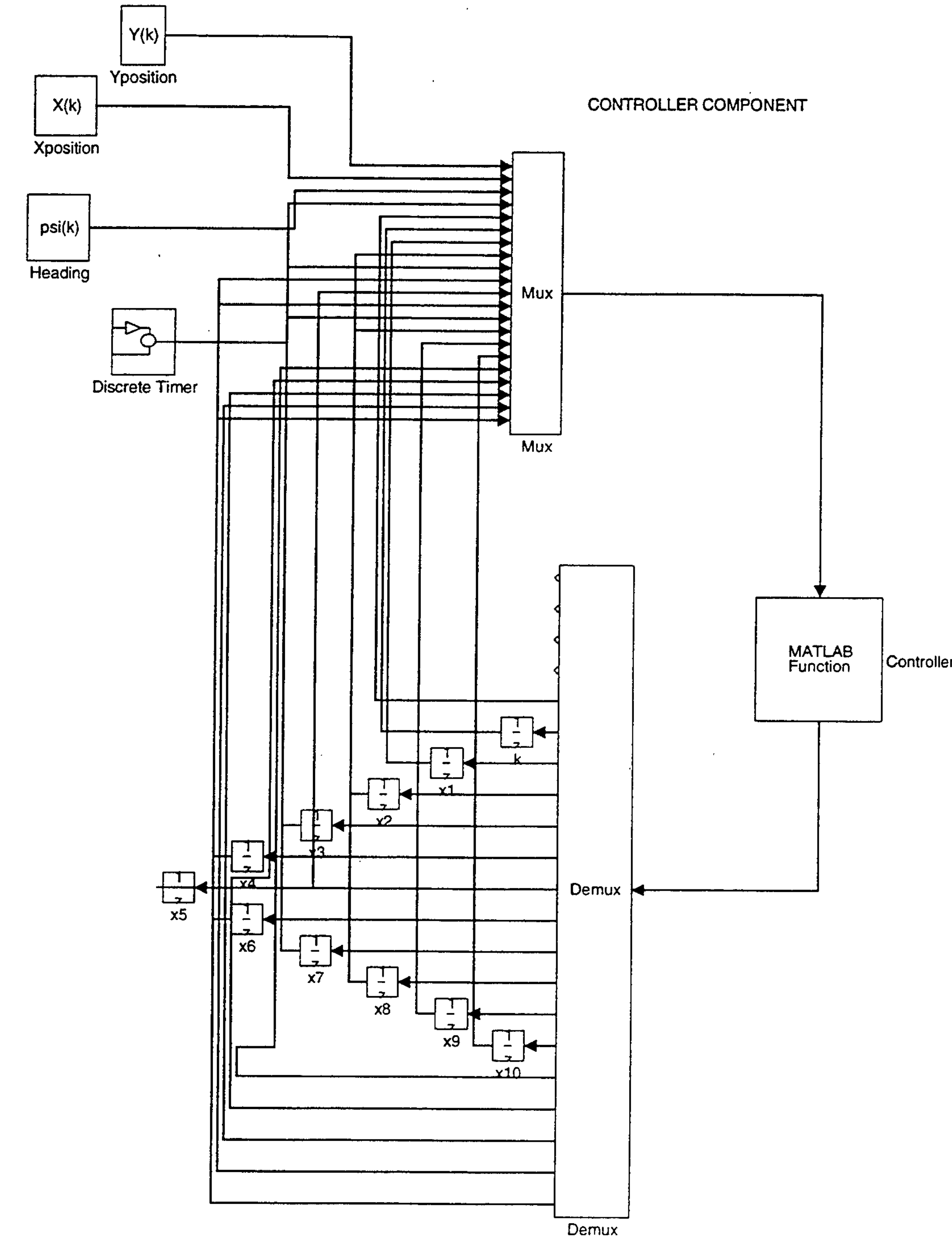
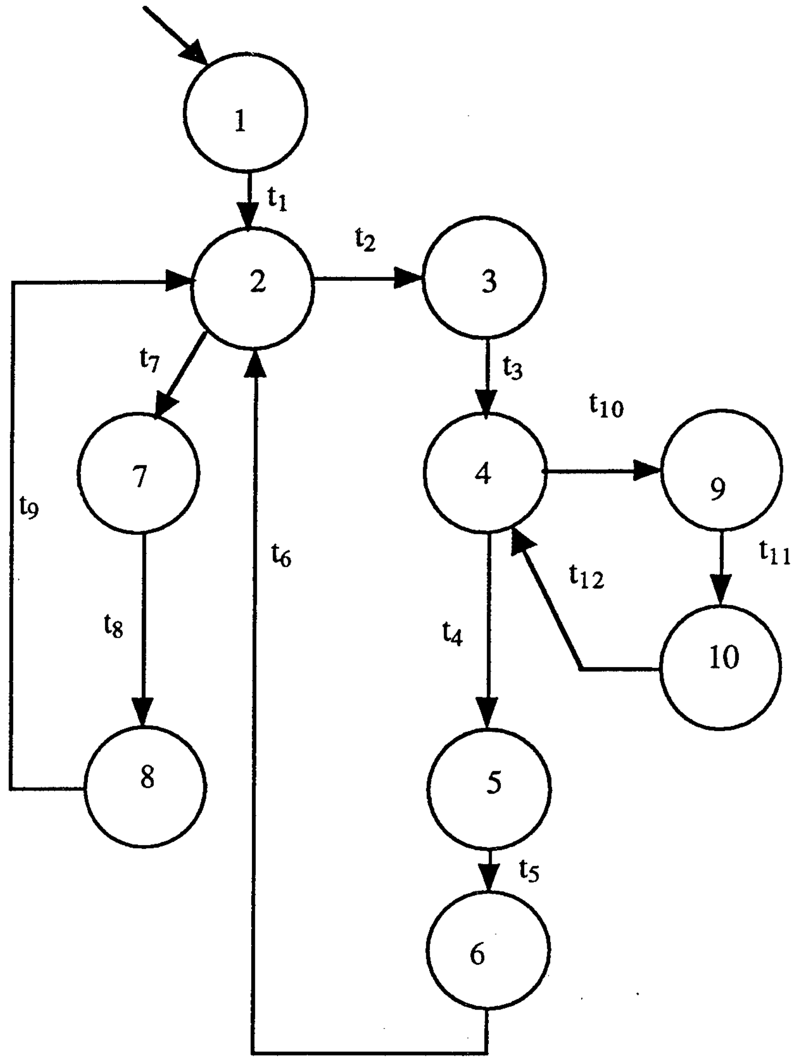


Figure 3.5 Block Diagram "Controller"




```
% Dynamic States
% state x1:  Get_First_UXO_Position
% state x2:  Nav_To_Position
% state x3:  Stop_Within_UXO_Proximity & Pick_Up_UXO
% state x4:  Carry_UXO_To_Disposal_Area
% state x5:  Stop & Drop_Off_UXO
% state x6:  Receive_New_UXO_Position
% State x7:  Obstacle_Detected_Enroute_To_UXO
% State x8:  Obstacle_Avoidance_Enroute_To_UXO
% State x9:  Obstacle_Detected_Enroute_To_Drop_Pile
% State x10: Obstacle_Avoidance_Enroute_To_Drop_Pile
```

```
% Transition States
% t1:   First_UXO_Position_Received
% t2:   Vehicle_Within_Range_Of_UXO
% t3:   UXO_Picked_Up
% t3:   UXO_Picked_Up
% t4:   Vehicle_Within_Range_Of_Drop_Pile
% t5:   Drop_Off_Complete
% t6:   New_UXO_Position_Received
% t7:   Obstacle_Detected_Enroute_To_UXO
% t8:   Signal_To_Do_Obstacle_Avoidance_Enroute_To_UXO
% t9:   Obstacle_Clear
% t10:  Obstacle_Detected_Enroute_To_Drop_Pile
% t11:  Signal_Do_Obstacle_Avoidance_Enroute_To_Drop_Pile
% t12:  Obstacle_Clear
```

Figure 3.6 State Space Logic Diagram "Petri Net"

IV. SIMULATIONS

A. SIMULATION SET UP AND PARAMETERS

The simulations conducted were designed to evaluate the effect of vehicle performance in UXO clearance operations under conditions of high slip (caused by rough terrain characteristics) and navigation error. Terrain characteristics such as topography, rocks, sand, mud, and other natural obstacles influence the vehicles ability to maintain a commanded course during UXO clearance operations. Slip will be modeled using a MATLAB uniformly distributed (0.0 to 1.0) random number generator. The code and associated block diagram configuration are shown in Appendix A. The slip coefficient (σ) is multiplied by the current track speed, then this quantity is subtracted from the velocity to give actual track speed over ground. Each track has it's own dedicated slip coefficient generator to model independent terrain variations. Current commercial GPS navigation systems provide navigation with 2 centimeter precision (perfect navigation in the context of this thesis). Navigation error is modeled by a normally distributed (0.0 to 1.0) random white noise generator from the SIMULINK source file. The white noise signal (error) is then passed through a first order "coloration" filter to provide temporal correlation characteristics where it is smoothed out and added to the current GPS location output to give sensed location. For simulation purposes, a 14 centimeter standard deviation navigation error which is realisically achieved in at least one differential gps system in use (Premier) is assumed. The block diagram configuration is shown in Appendix A.

The following equations will be used in the simulation:

Track # 1 Slip Velocity $vs1 = v1(k) - v1(k)*\sigma 1$

Track # 2 Slip Coefficient $vs2 = v2(k) - v2(k)*\sigma 2$

$X[(k+1)\ dt] = [exp(dt)]*X[k\ dt] + [1\text{-}exp(dt/\tau)]*\mu 1[k\ dt)$

$Y[(k+1)\ dt] = [exp(dt/\tau)]*Y[k\ dt] + [1 - exp(dt/\tau)]*\mu 2[k\ dt)$

where, σ = slip coefficient

$dt = 1$

$\tau = 106$

μ = navigation error noise coefficient

B. SIMULATION SCENARIOS

1. No Slip with Perfect Navigation

a. Path Deviation Test (Point to Point)

This simulation has been conducted for target distances of 10, 20, and 30 meters. Starting from position (0,0), the vehicle is sent to a target 10 meters away to position (10,10). This scenario, because of the assumed randomness in errors and slippage, was run 10 times, with each scenario path recorded and stored in a data file. The 10 paths were then super-imposed and plotted against the ideal path on a XY graph. The lateral path deviation was calculated from the recorded path data using MATLAB. The following data and equations will be used to compute standard deviation.

x( ) = x path coordinates for run ( )

y( ) = y path coordinates for run ( )

$Xi = [x1; x2; x3; x4; x5; x6; x7; x8; x9; x10]$

$Yi = [y1; y2; y3; y4; y5; y6; y7; y8; y9; y10]$

$\sum i = 1/\sqrt{2} * Xi - 1/\sqrt{2} * Yi$

$\sigma(dev) = std(\sum i)$ %[standard deviation of $\sum$i (MATLAB function)]

b. Clearance Test

For this scenario, five pre-designated UXO targets were programmed into the vehicle. Starting from location (0,0), the vehicle assignment was to go to the UXO, perform a pickup maneuver, and carry it away to the disposal location (0,0). This was done until all UXOs were cleared from the field. A time history was recorded and the vehicle path plotted.

2.No Slip with Navigation Error

a. Path Deviation Test (Point to Point)

This simulation was conducted for target distances of 10, 20, and 30 meters. Starting from position (0,0), the vehicle was sent to a target 10 meters away to position (10,10). This scenario was run 10 times, with each scenario path recorded and stored in a data file. The 10 paths were super-imposed and plotted against the ideal path on a XY graph. The lateral path deviation was calculated from the recorded path data using MATLAB.

b. Clearance Test

For this scenario, five pre-designated UXO targets were programmed into the vehicle. Starting from location (0,0), the vehicle assignment was to go to the UXO.

perform a pickup maneuver, and carry it away to the disposal location (0,0). This was done until all UXOs were cleared from the field.

3. Slip with Perfect Navigation

a. Path Deviation Test (Point to Point)

This simulation was conducted for target distances of 10, 20, and 30 meters. Starting from position (0,0), the vehicle was sent to a target 10 meters away to position (10,10). This scenario was run 10 times, with each scenario path recorded and stored in a data file.

b. Clearance Test

For this scenario, five pre-designated UXO targets will be programmed into the vehicle. Starting from location (0,0), the vehicle assignment will be to go to the UXO, perform a pickup maneuver, and carry it away to the disposal location (0.0). This will be done until all UXOs have been cleared from the field. A time history will be recorded and the vehicle path will be plotted on an XY graph.

4. Slip with Navigation Error

The identical scenario with both wheel slip and navigational error was run. Each path was recorded and super-imposed to show the path randomness caused by slip and navigation error. These result were compared and analyzed for path deviation and clearance time.

V. DISCUSSION OF SIMULATION RESULTS

A. NO WHEEL SLIP WITH PERFECT NAVIGATION

1. Clearance Time (Ordnance Clearing Scenario)

Five pre-assigned UXO targets were programmed into the vehicle model. The vehicle was then sent out to clear the field while time and vehicle path were recorded. As expected, path deviation was non-existent during this scenario. Figure 5.1 shows the vehicle path for this simulation. The time it took to find and clear all five assigned targets was 18.7 minutes.

B. NO WHEEL SLIP WITH NAVIGATION ERROR

1. Path Deviation (Point to Point Transit)

The point to point navigation test was conducted to examine vehicle path lateral deviation due to a 14 centimeter navigation error. Tests were conducted at 10, 20, 30 meters. At each distance, 10 scenarios were run and their respective path tracks stored in a data file. The 10 tracks were then super-imposed and plotted against the ideal path as indicated with asterisks. On the basis of the ten experiments, mean and standard deviation cross track errors were computed and the following results were calculated:

Distance (meters)	Maximum Lateral Path Deviation (meters)
10	0.1499
20	0.1149
30	0.0975

Path deviation due to navigation error alone was very low. As distance increased, the

path deviation decreased. Even at 10 meters, the deviation was within the 14 centimeter navigation error range. These results are shown in Figures 5.2, 5.3, and 5.4 respectively.

2. Clearance Time (Ordnance Clearing Scenario)

Five pre-assigned UXO targets were programmed into the vehicle model. The vehicle was then sent out to clear the field while time and vehicle path were recorded. As expected, path deviation was non-existent during this scenario. Figure 5.5 shows the vehicle path for this simulation. Clearance time was 18.83 minutes. This took just 0.13 minutes longer than the no slip-perfect navigation case, indicating that the vehicle control system accomodates navigational errors well.

C. WHEEL SLIP WITH PERFECT NAVIGATION

1. Path Deviation (Point to Point Transit)

The point to point navigation test was conducted to examine vehicle path lateral deviation due to wheel slip alone. Tests were conducted at 10, 20, 30 meters. At each distance, 10 scenarios were run and their respective path tracks stored in a data file. The 10 tracks were then super-imposed and plotted against the ideal path as indicated with asterisks. The following results were calculated:

Distance (meters)	Maximum Lateral Path Deviation (meters)
10	0.5084
20	0.6115
30	0.7451

Path deviation due to slip alone was considerably higher than the two previous cases.

As path distance increased from 10 to 30 meters, lateral deviation increase by almost a factor of 0.1 meters. The results are shown in figures 5.6, 5.7, and 5.8 respectively.

2. Clearance Time (Ordnance Clearing Scenario)

Five pre-assigned UXO targets were programmed into the vehicle model. The vehicle was then sent out to clear the field while time and vehicle path were recorded. As expected, path deviation was non-existent during this scenario. Figure 5.9 shows the vehicle path for this simulation. Clearance time was 37 minutes. This is about twice the time it took for the no slip-navigation error case. As can be seen, clearance time was greatly effected by the introduced slip - mostly because of the effect on average forward speed reduction.

D. WHEEL SLIP WITH NAVIGATION ERROR

1. Path Deviation (Point to Point Transit)

The point to point navigation test was conducted to examine vehicle path lateral deviation due to wheel slip and a 14 centimeter navigation error. Tests were conducted at 10, 20, 30 meters. At each distance, 10 scenarios were run and their respective path tracks stored in a data file. The 10 tracks were then super-imposed and plotted against the ideal path as indicated with asterisks. The following results were calculated:

Distance (meters)	Maximum Lateral Path Deviation (meters)
10	0.4208
20	0.6093
30	1.2

Path deviation due to slip alone was considerably higher than the two previous cases. As path distance increased from 10 to 30 meters, lateral deviation increased. These results are very consistent with the slip-perfect navigation case. The results are shown in figures 5.10, 5.11, and 5.12 respectively.

2. Clearance Time (Ordnance Clearing Scenario)

Five pre-assigned UXO targets were programmed into the vehicle model. The vehicle was then sent out to clear the field while time and vehicle path were recorded. As expected, path deviation was non-existent during this scenario. Figure 5.13 shows the vehicle path for this simulation. Clearance time was 37.3 minutes. This took just 0.3 minutes longer than the no slip-perfect navigation case. The introduction of navigation error has minimal effect on clearance time.

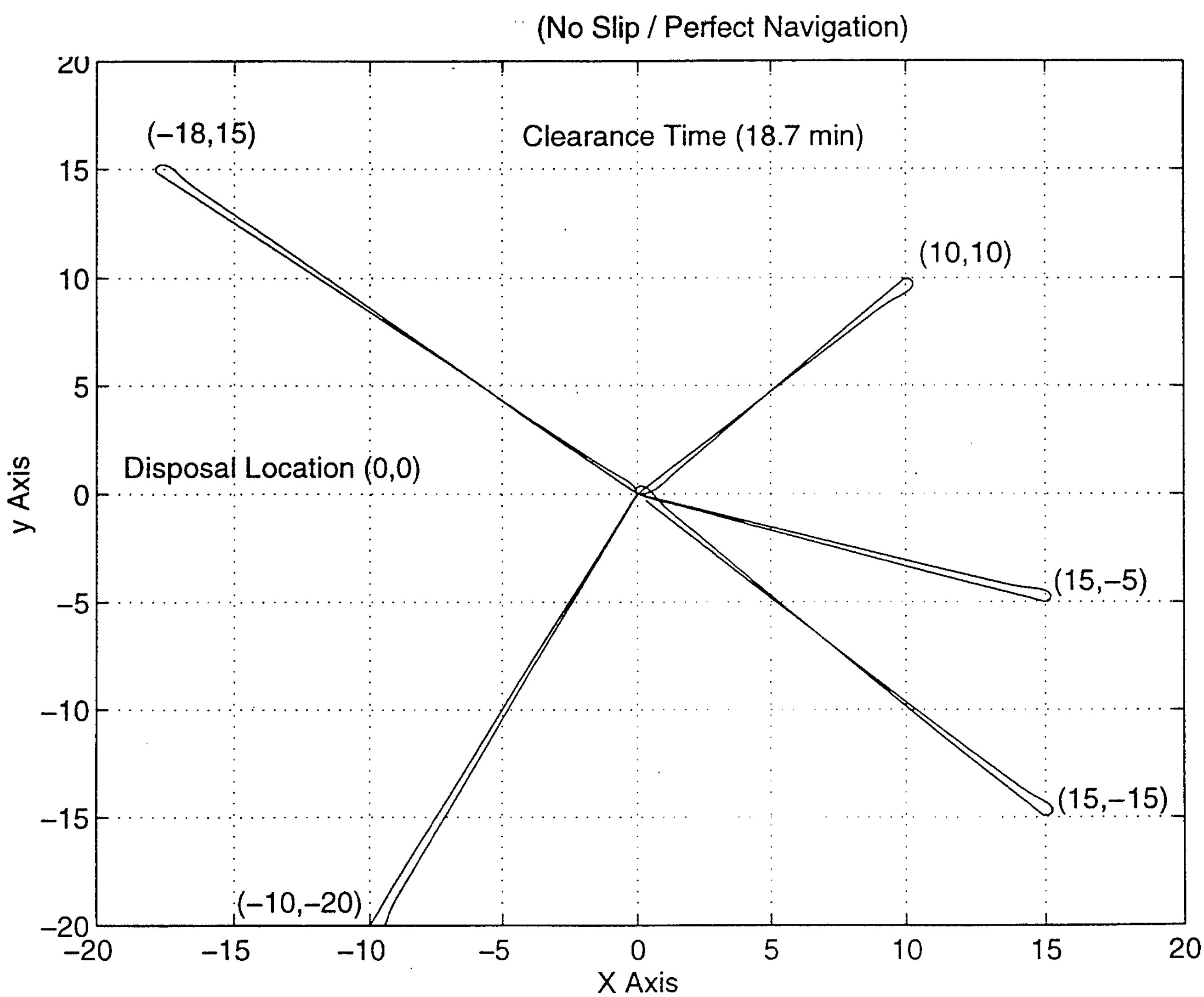


Figure 5.1 No Slip with Perfect Navigation

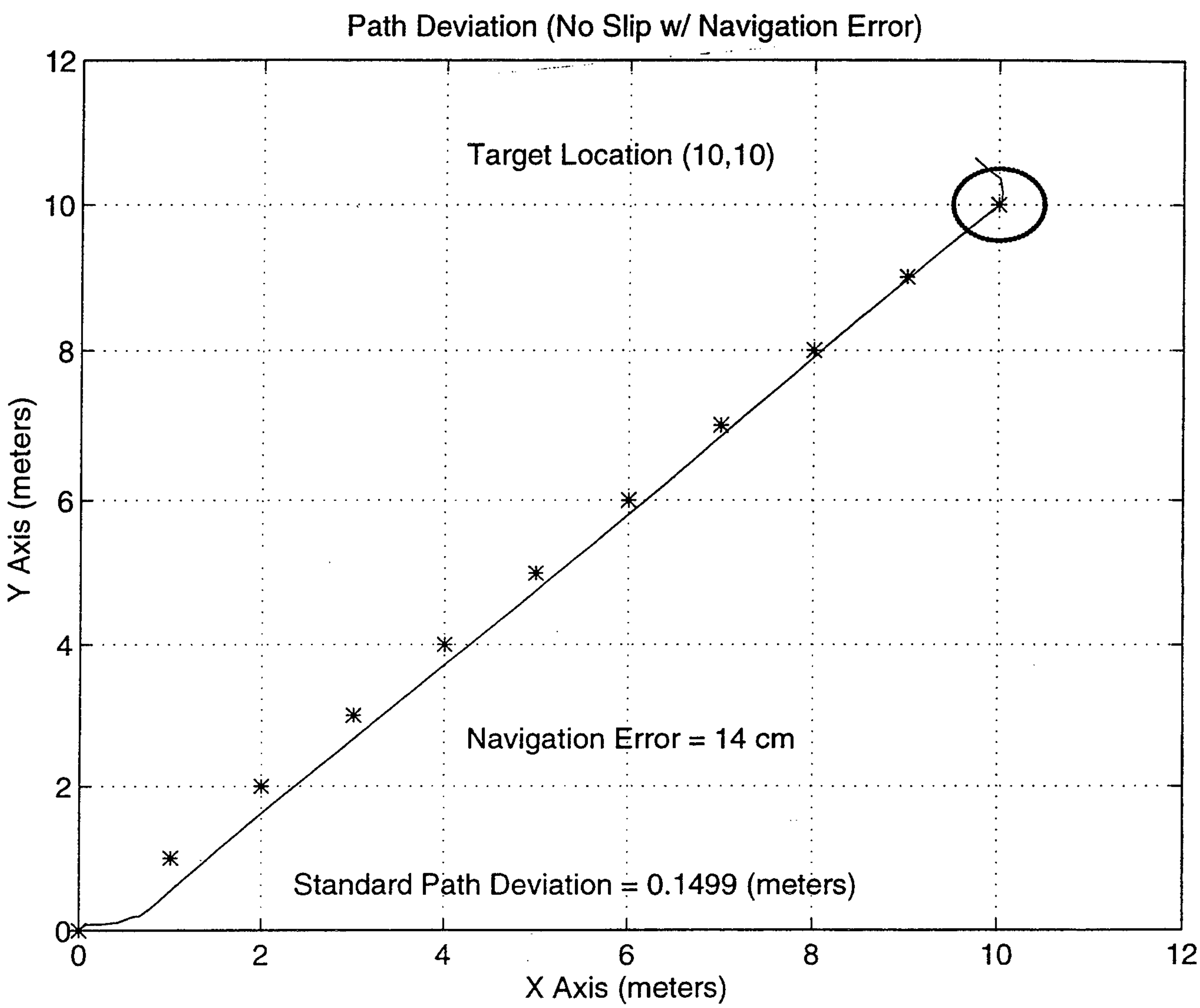


Figure 5.2 Path Deviation for 10 meters "No Slip with Navigation Error"

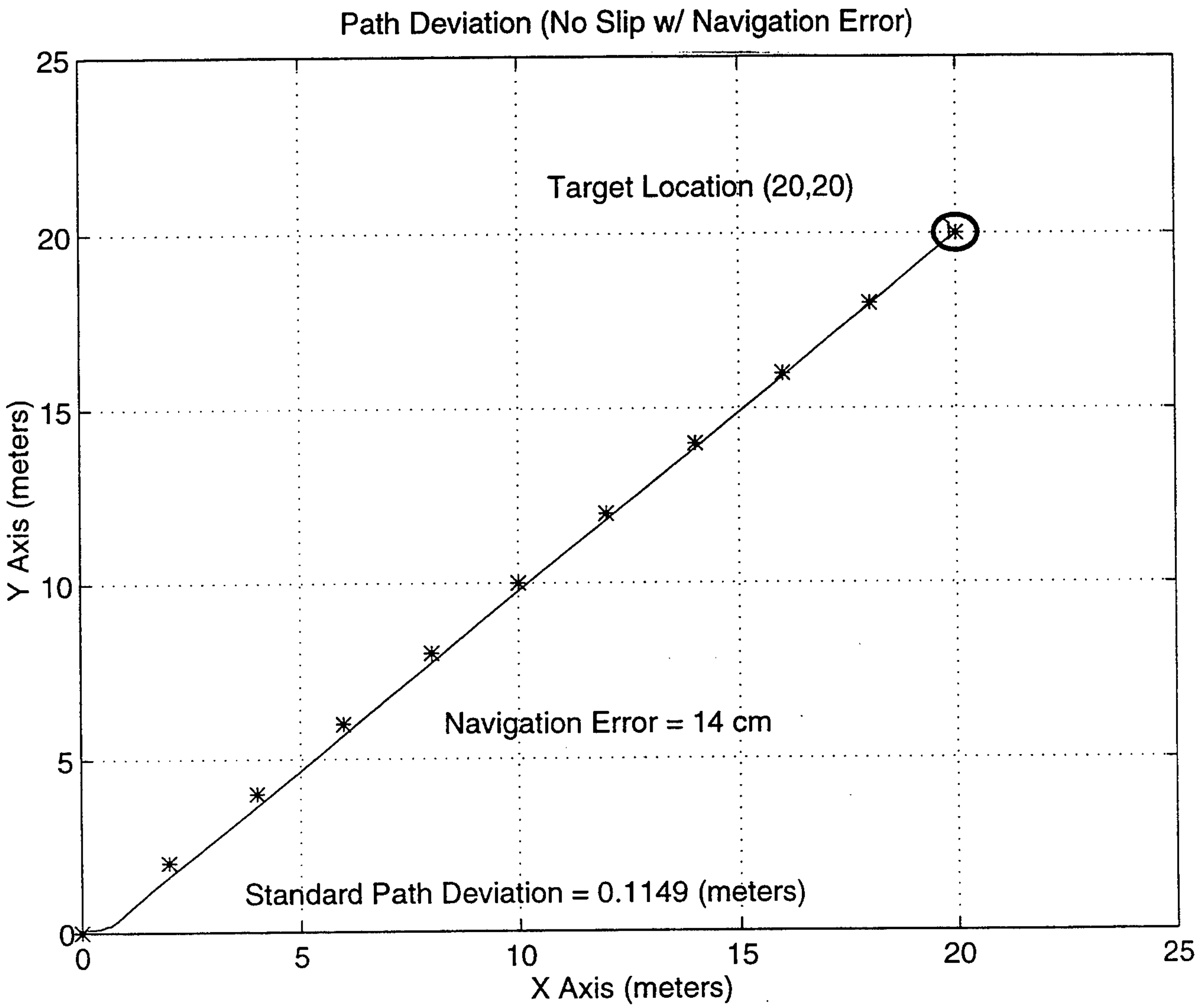


Figure 5.3 Path Deviation for 20 meters "No Slip with Navigation Error"

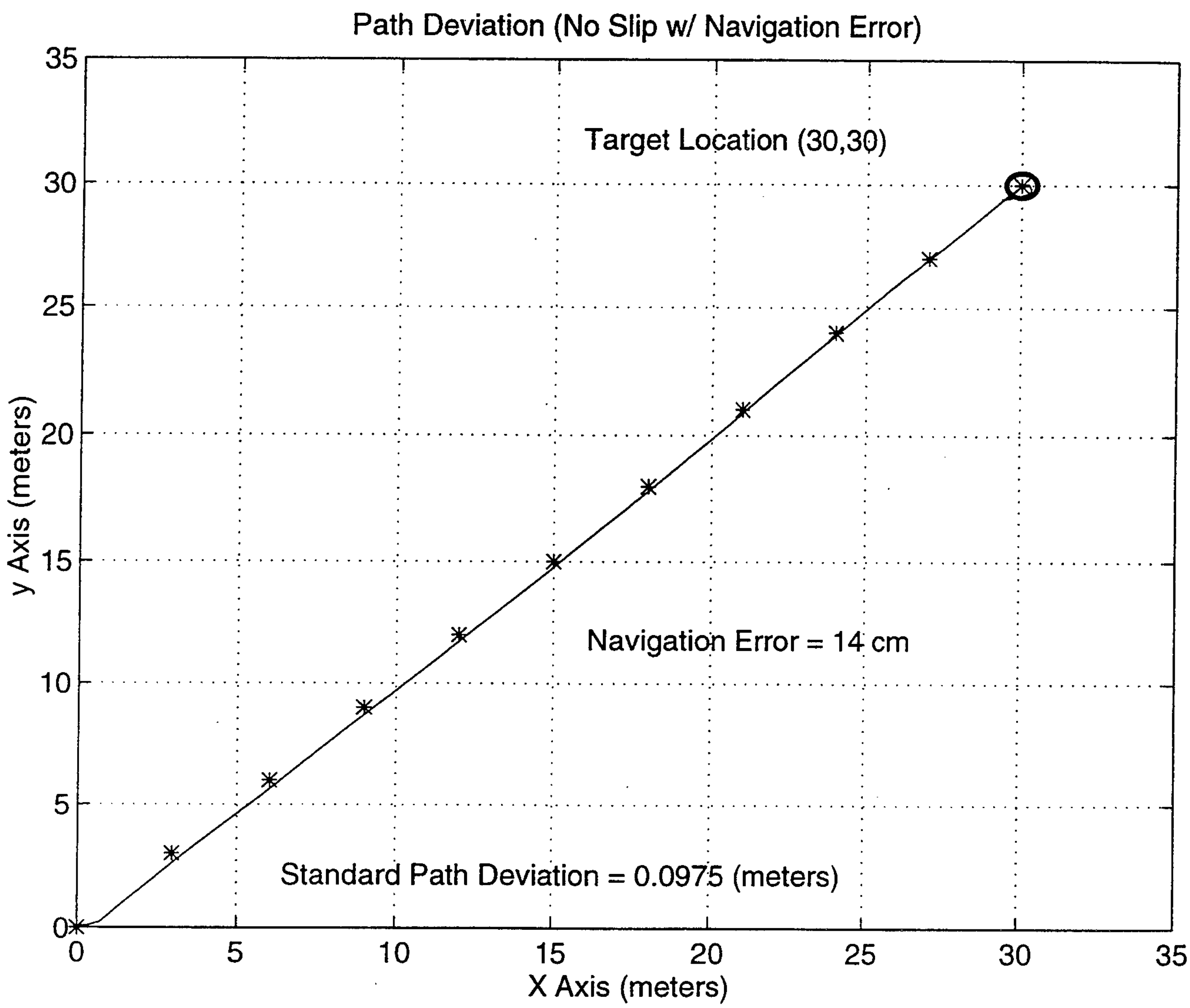


Figure 5.4 Path Deviation for 30 meters "No Slip with Navigation Error"

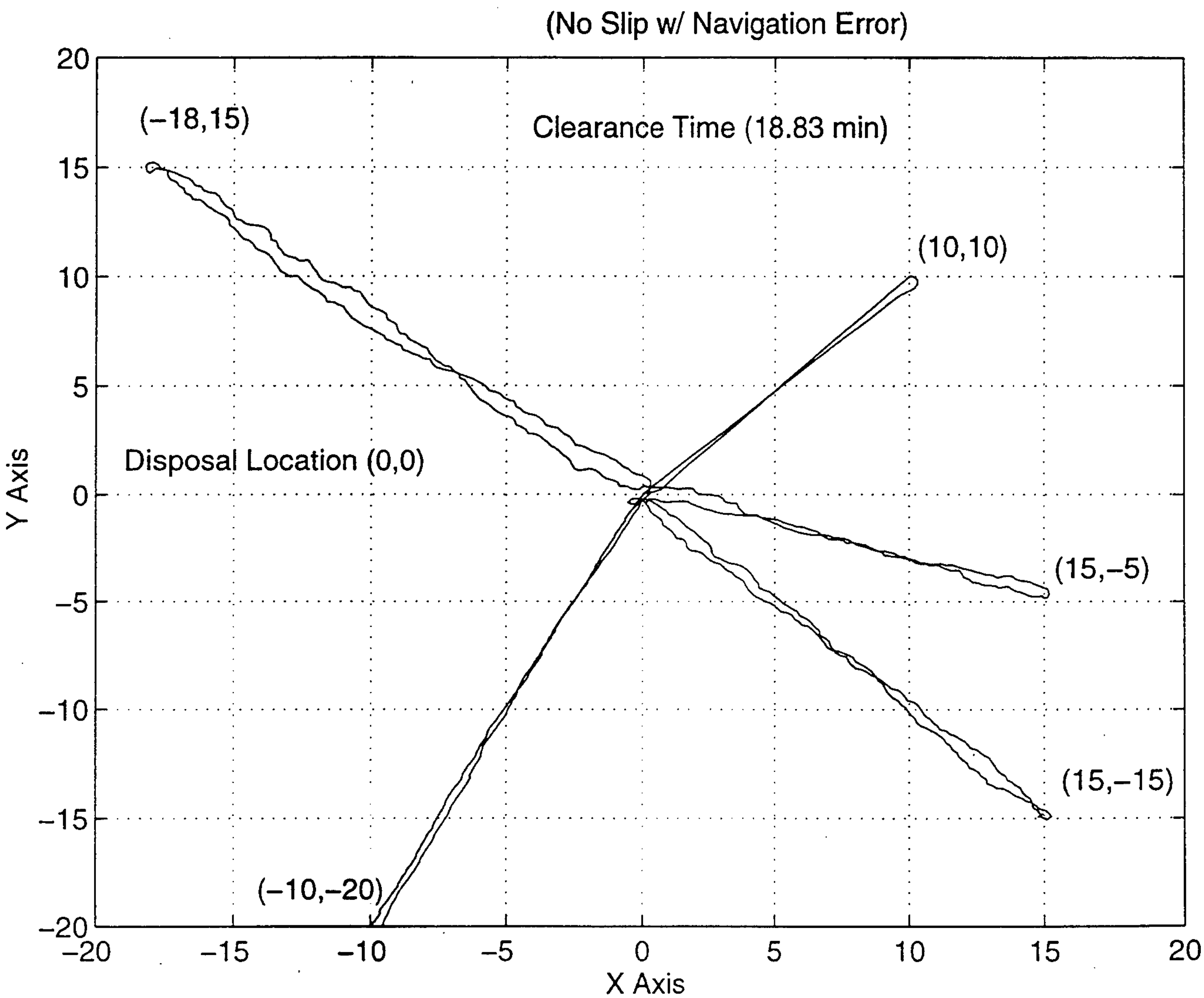


Figure 5.5 UXO Clearance Scenario "No Slip with Navigation Error"

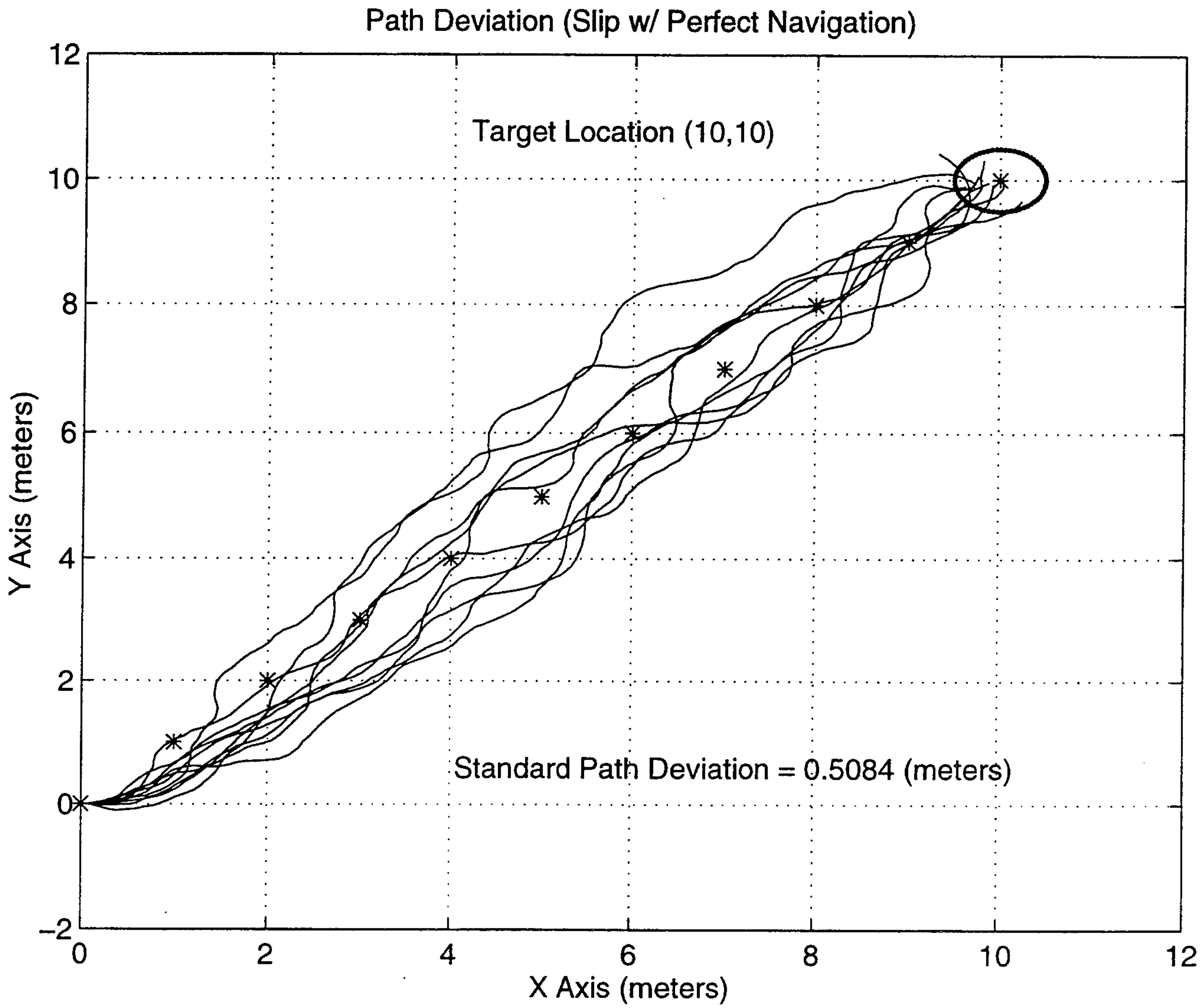


Figure 5.6 Path Deviation for 10 meters "Slip with Perfect Navigation"

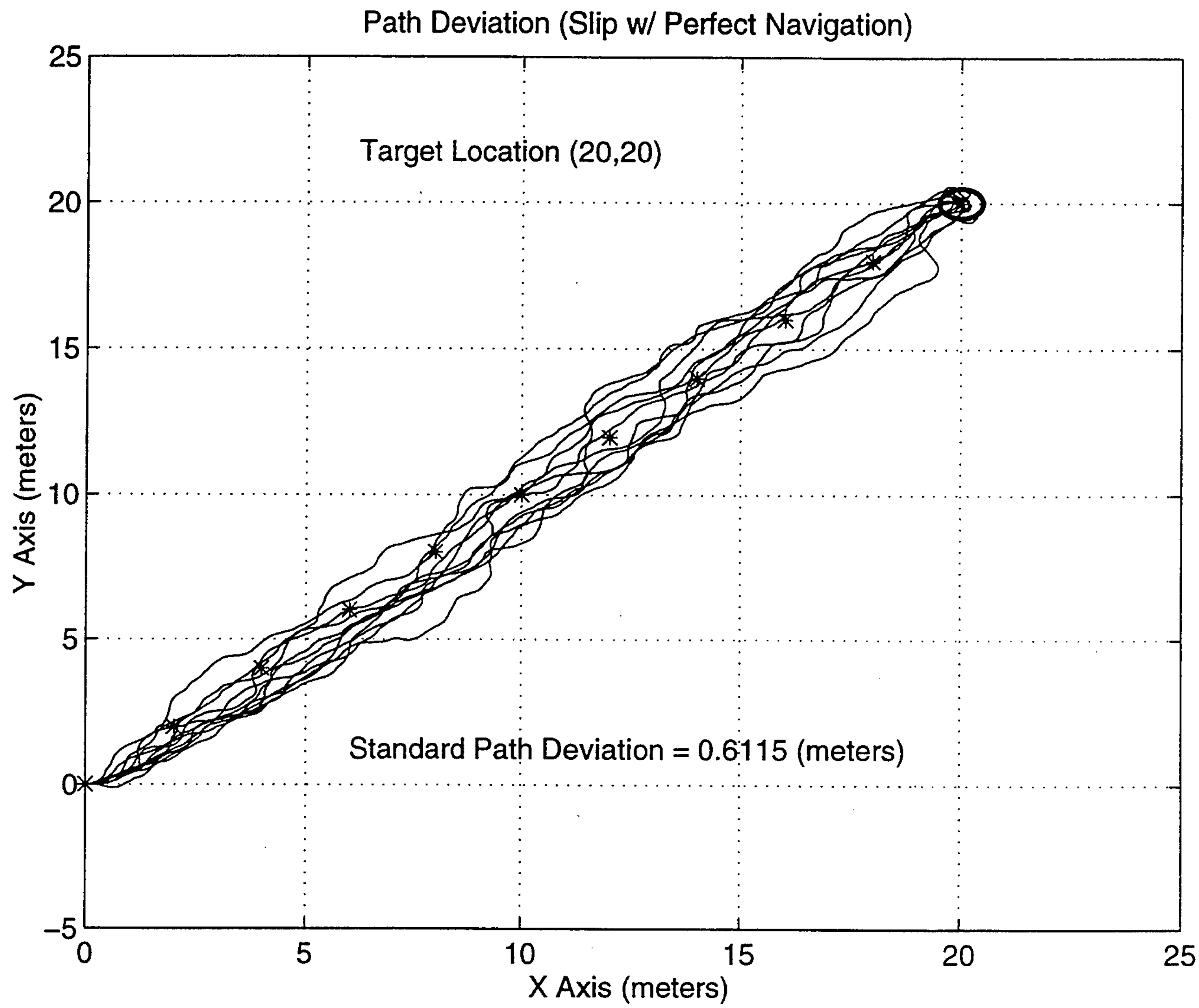


Figure 5.7 Path Deviation for 20 meters "Slip with Perfect Navigation"

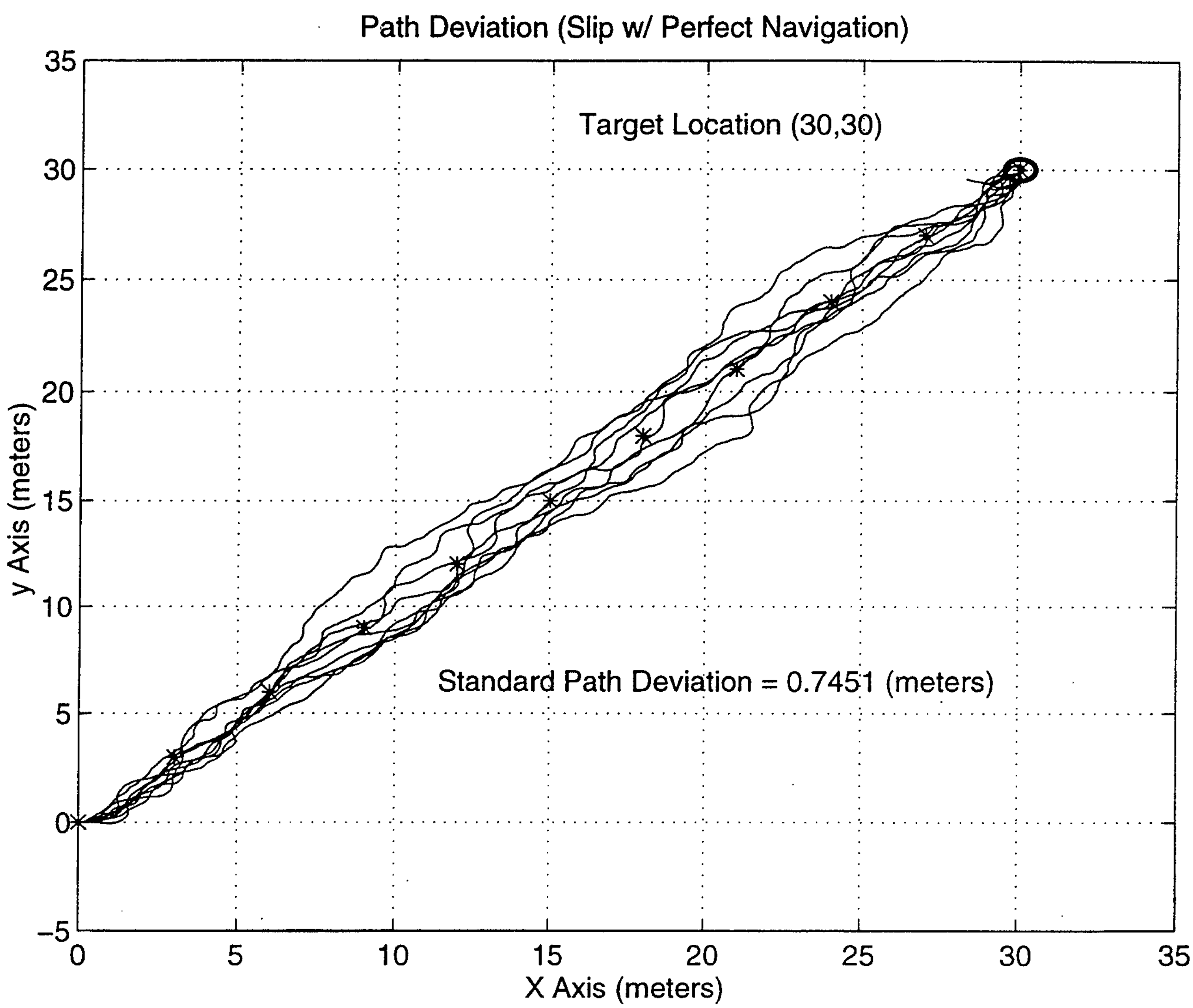


Figure 5.8 Path Deviation for 30 meters "Slip with Perfect Navigation"

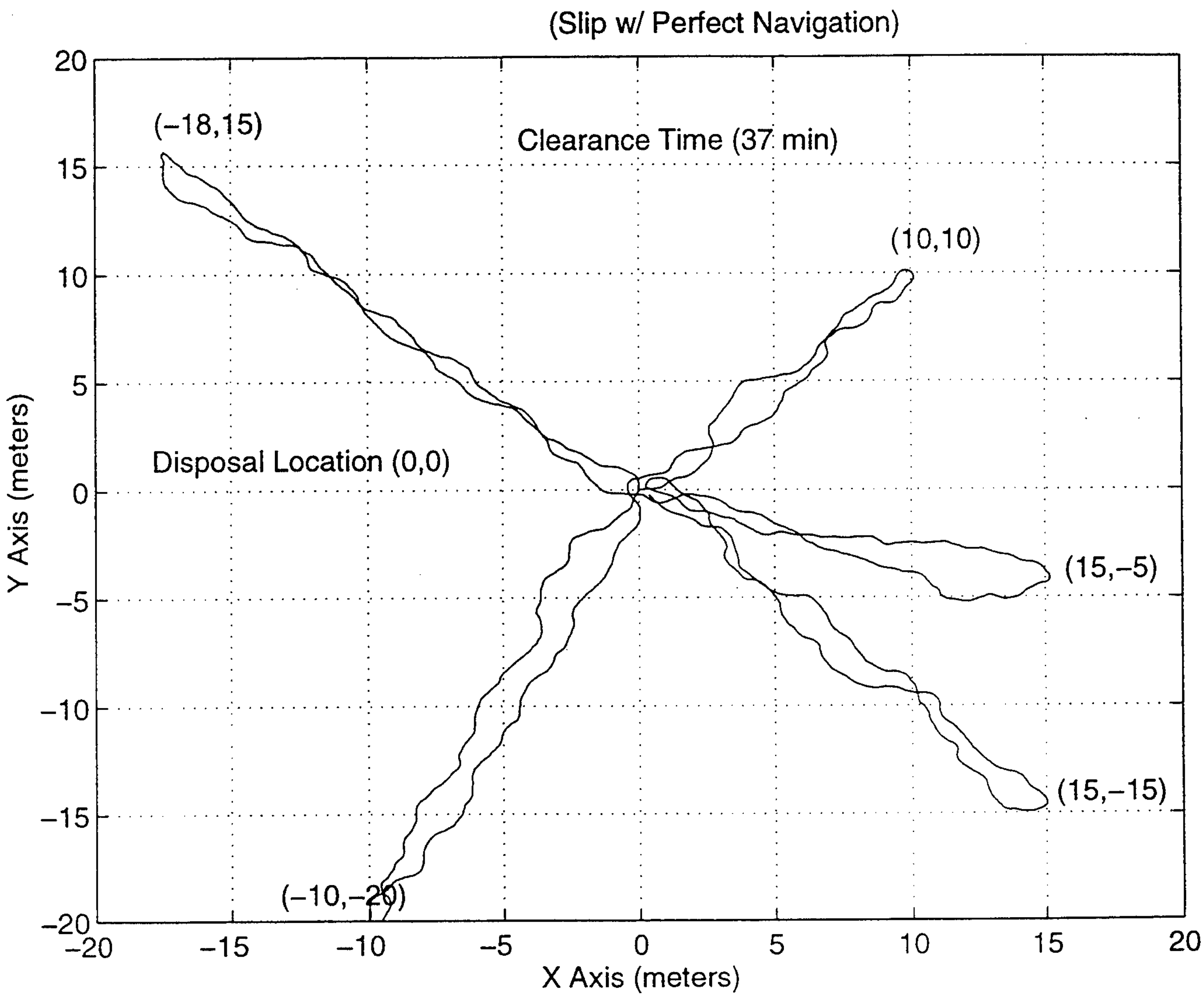


Figure 5.9 UXO Clearance Scenario "Slip with Perfect Navigation"

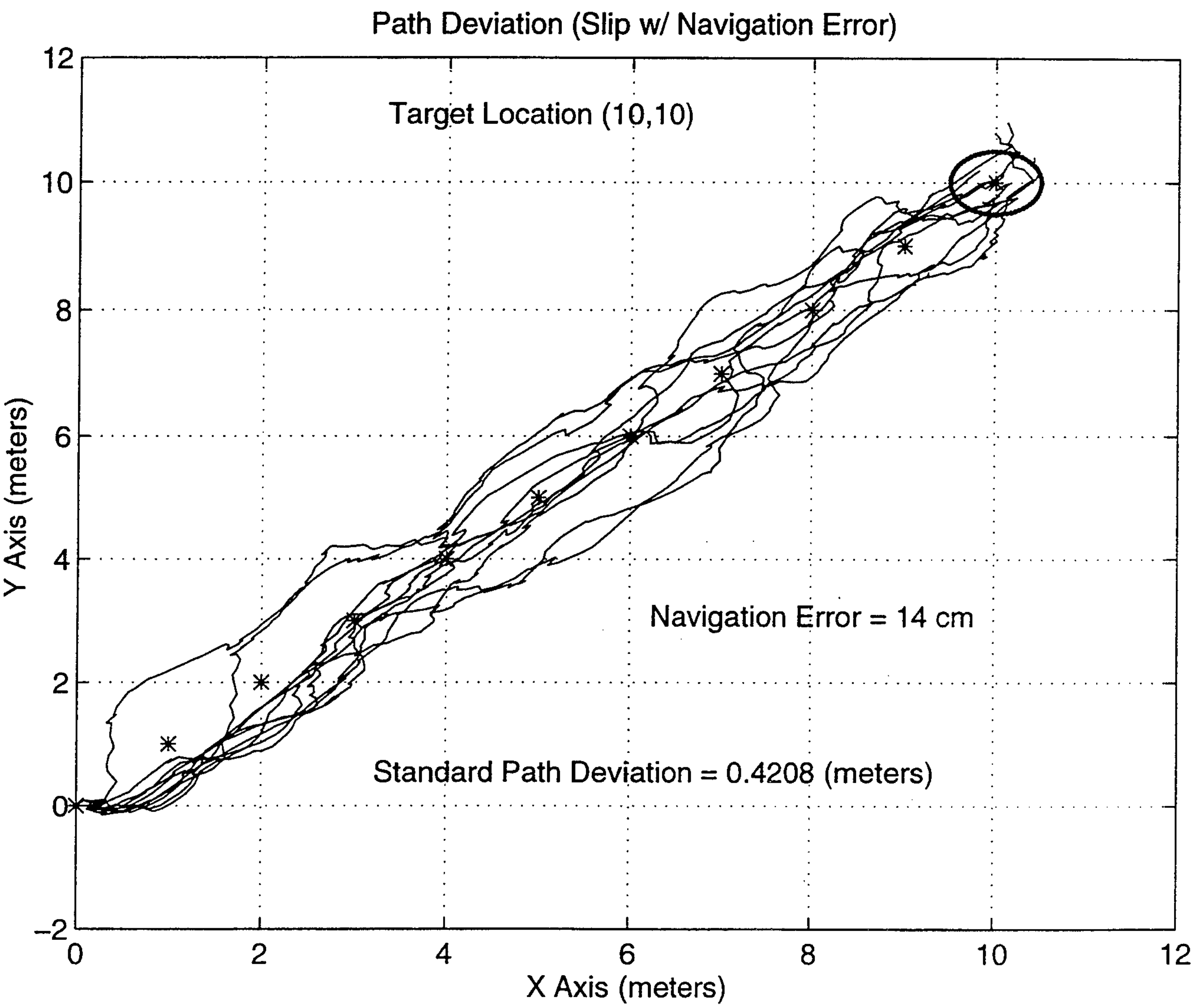


Figure 5.10 Path Deviation for 10 meters "Slip with Navigation Error"

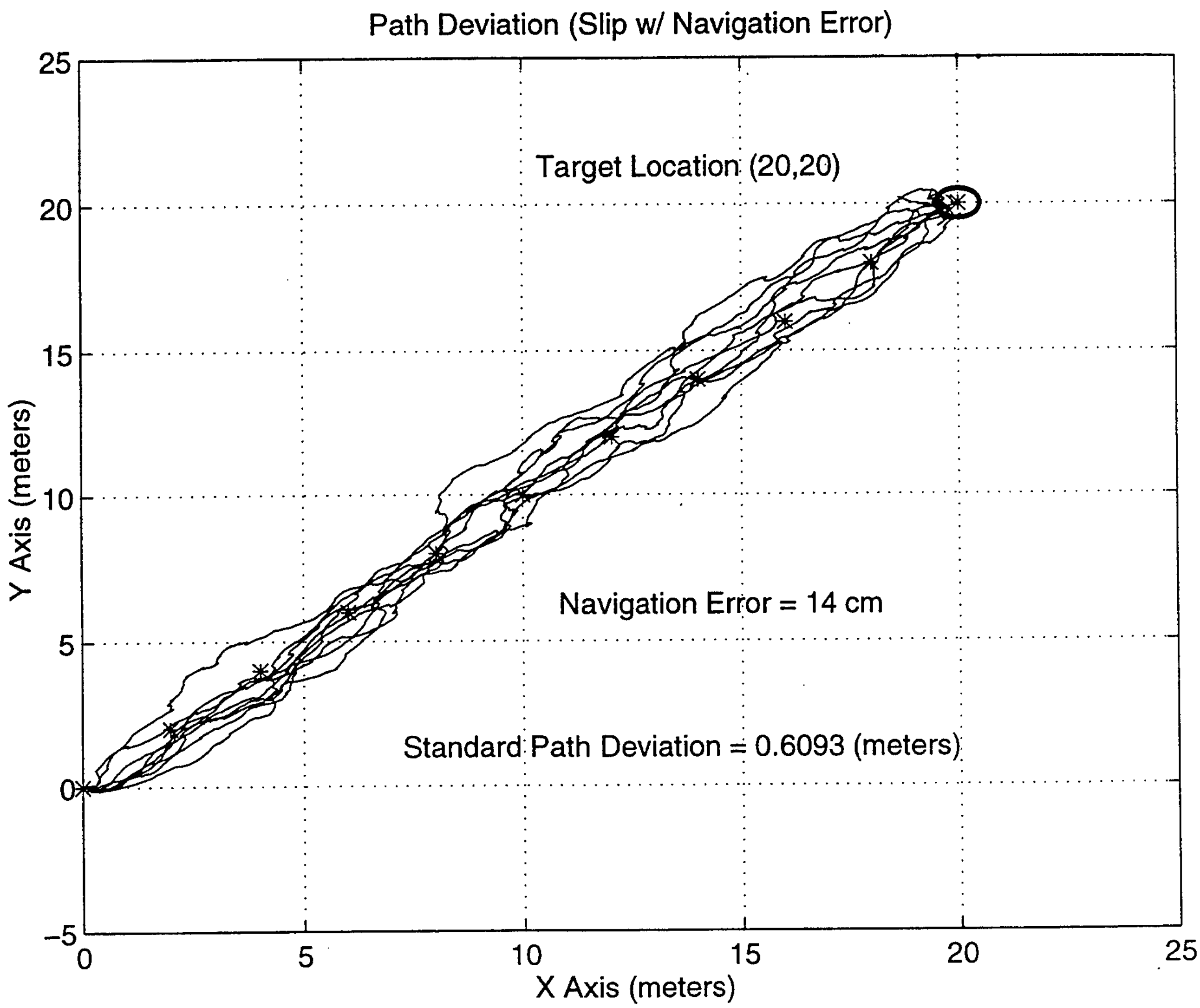


Figure 5.11 Path Deviation for 20 meters "Slip with Navigation Error"

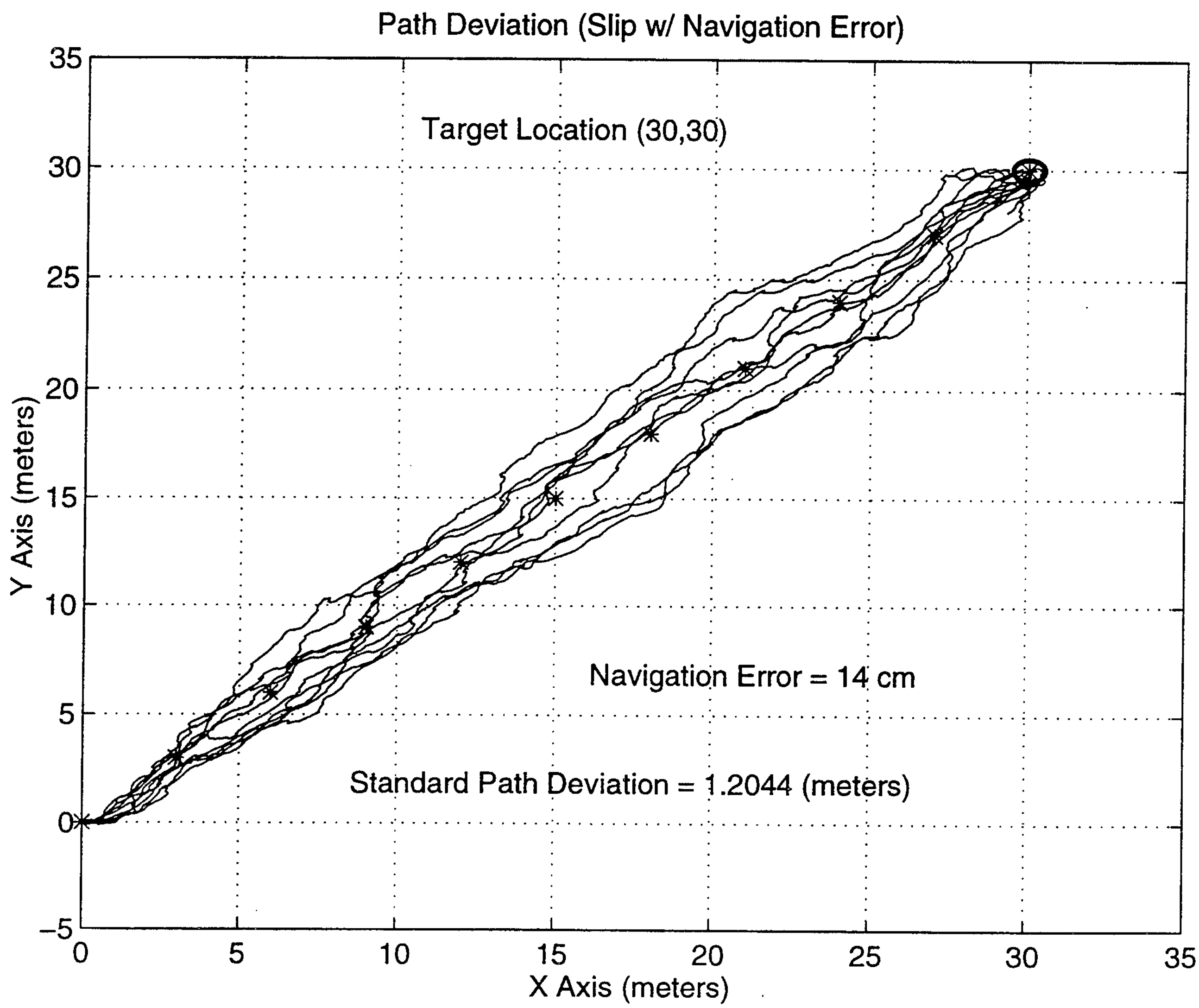


Figure 5.12 Path Deviation for 30 meters "Slip with Navigation Error"

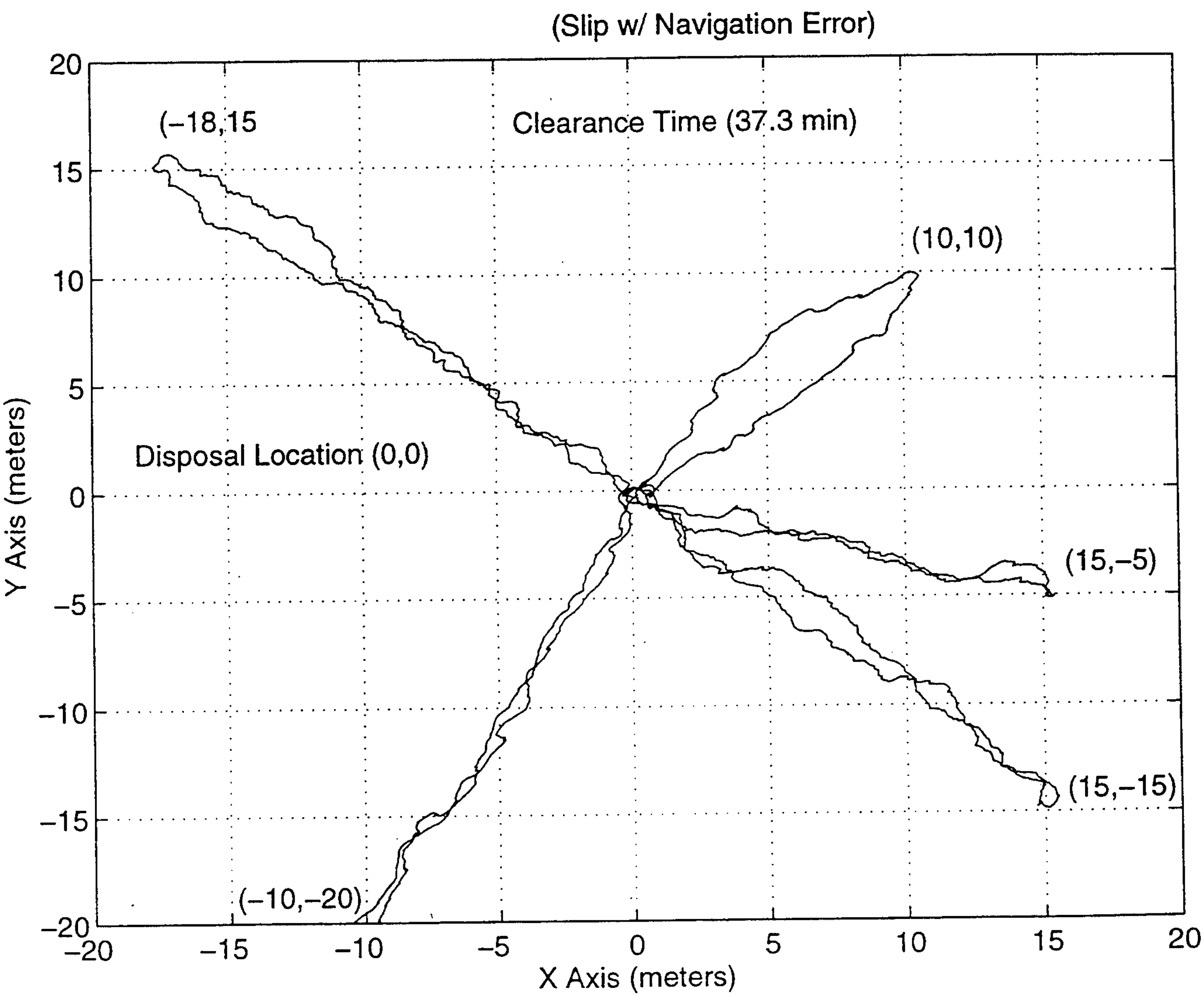


Figure 5.13 UXO Clearance Scenario "Slip with Navigation Error"

VI. CONCLUSIONS

The primary purpose of this thesis was to develop a computer model to simulate the dynamic and kinematic characteristics of a small tracked robotic vehicle used for unexploded ordnance (UXO) gathering operations. After model development, the secondary purpose was to examine vehicle performance in UXO gathering operations under the influence of track slip (due to terrain characteristics) and navigational error. It was found that vehicle performance could reasonably be modeled using a discrete time / discrete event state space control scheme. Using SIMULINK as a modeling tool, discrete time state space equations can be modeled with block diagrams. Discrete state modeling including the state transition logic is built into a separate file, linked to the main system's blocks through a multiplexing / demultiplexing of the discrete state signals. Multiple point to point scenarios run under varying conditions (ie; no slip-perfect navigation, no slip-navigation error, slip-perfect navigation, and slip-navigation error) demonstrated that vehicle performance was significantly reduced by track slip more than navigation error. Track slip caused a greater path deviation than did navigation error. In some cases, track slip caused the vehicle to deviate from it's path more than twice the distance caused by realistic navigation error. UXO clearance scenarios also demonstrated track slip caused more of a degradation in performance than did navigation error. UXO clearance time doubled when track slip is introduced. Navigation error alone had minimal effect on clearance time compared to ideal conditions of no slip with perfect navigation. The speed of solution with SIMULINK is too slow to be generally useful in complex scenarios so one recommendation is that the auto-

code generation capabilities of the simulation software be used to get a C code version that would run many times faster. Otherwise the techniques using SIMULINK as explained here are convenient.

APPENDIX . SYSTEM DIAGRAMS AND MODELING EQUATIONS

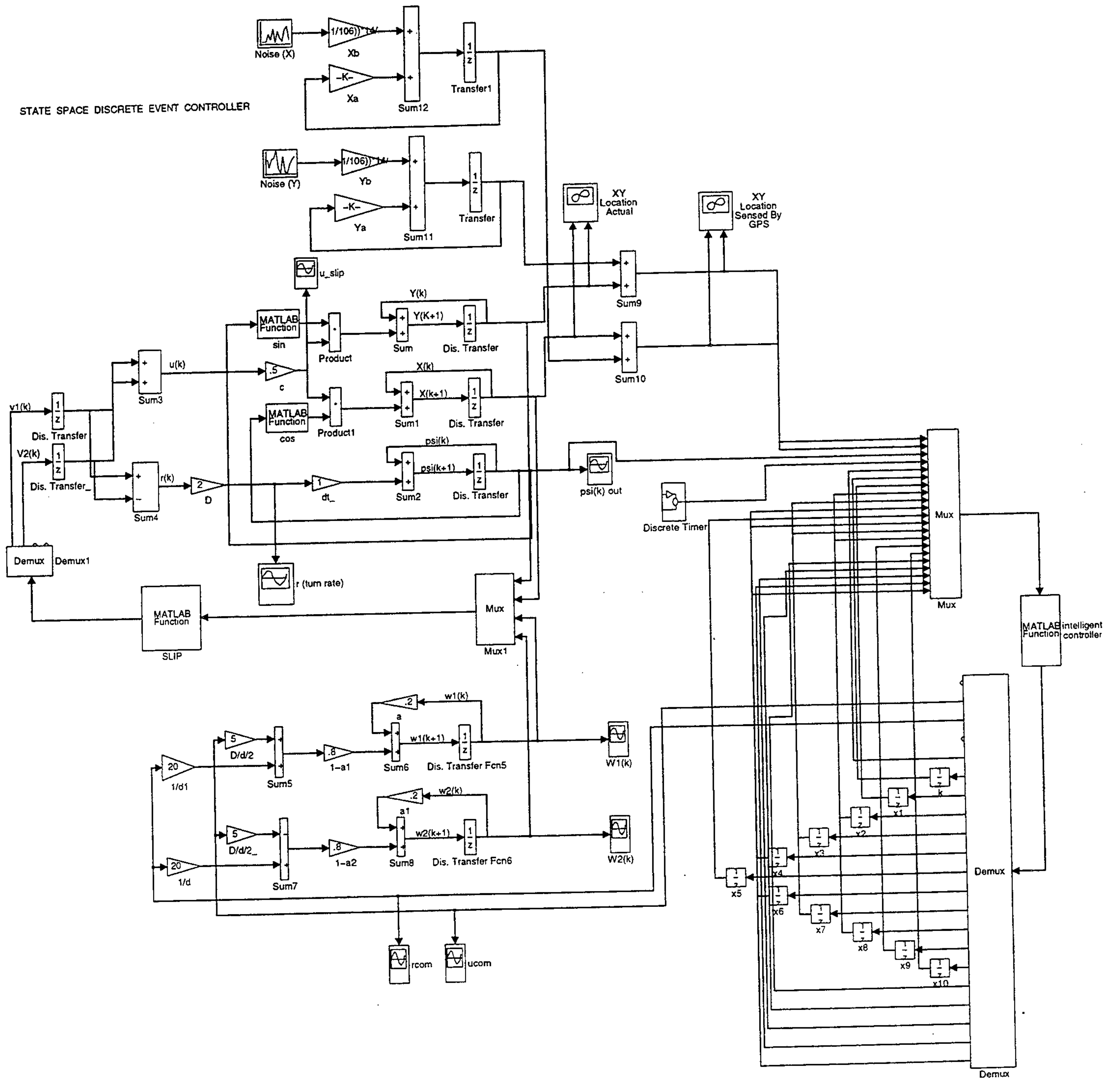
This Appendix contains the derived state space equations used to model vehicle kinematic and dynamic characteristics. Also included is a full simulink system model emplementing these equations with associated source codes.

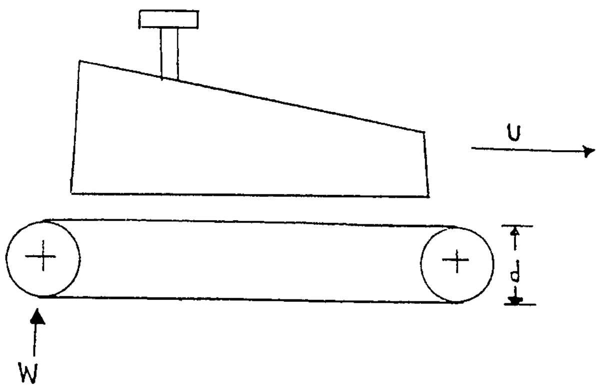
```
% BUG Controller

% Dynamic States
% state x1:  Get_First_UXO_Position
% state x2:  Nav_To_Position
% state x3:  Stop_Within_UXO_Proximity & Pick_Up_UXO
% state x4:  Carry_UXO_To_Disposal_Area
% state x5:  Stop & Drop_Off_UXO
% state x6:  Receive_New_UXO_Position
% State x7:  Obstacle_Detected_Enroute_To_UXO
% State x8:  Obstacle_Avoidance_Enroute_To_UXO
% State x9:  Obstacle_Detected_Enroute_To_Drop_Pile
% State x10: Obstacle_Avoidance_Enroute_To_Drop_Pile

% Transition States
% t1:    First_UXO_Position_Received
% t2:    Vehicle_Within_Range_Of_UXO
% t3:    UXO_Picked_Up
% t3:    UXO_Picked_Up
% t4:    Vehicle_Within_Range_Of_Drop_Pile
% t5:    Drop_Off_Complete
% t6:    New_UXO_Position_Received
% t7:    Obstacle_Detected_Enroute_To_UXO
% t8:    Signal_To_Do_Obstacle_Avoidance_Enroute_To_UXO
% t9:    Obstacle_Clear
% t10:   Obstacle_Detected_Enroute_To_Drop_Pile
% t11:   Signal_Do_Obstacle_Avoidance_Enroute_To_Drop_Pile
% t12:   Obstacle_Clear


function [y]= bugstate(x)

% Initial Conditions
X=0;
Y=0;
psicom=0;
psi=0;
rcom=0;
ucom=0;
k=0;
t03=0;
t05=0;
t06=0;
t08=0;
t010=0;
t=0;
x1=0;x2=0;x3=0;x4=0;x5=0;x6=0;x7=0;x8=0;x9=0;x10=0;
t1=0;t2=0;t3=0;t4=0;t5=0;t6=0;t7=0;t8=0;t9=0;t10=0;t11=0;t12=0;

% Input Vector y[x] Component Designation
Y=x(1)
X=x(2)
psi=x(3)
t=x(4)
t0=x(5)
k=x(6)
x1=x(7)
x2=x(8)
x3=x(9)
x4=x(10)
x5=x(11)
x6=x(12)
x7=x(13)
x8=x(14)
x9=x(15)
x10=x(16)
t03=x(17)
t05=x(18)
t06=x(19)
t08=x(20)
t010=x(21)

target1
obstacles
```

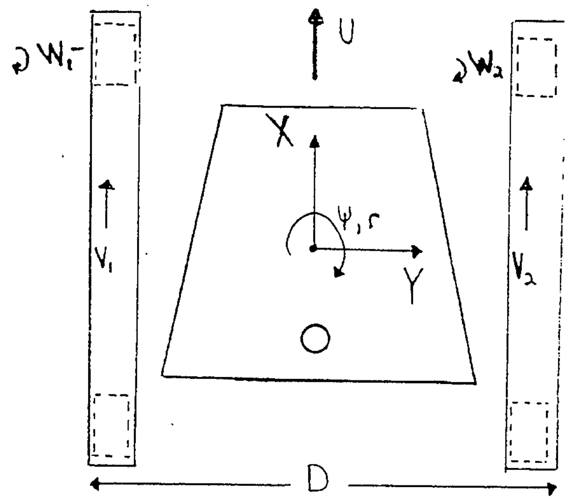
```
        if psi>2*pi
                psi=psi-2*pi;
        elseif psi<-2*pi
                psi=psi+2*pi;
        end
        end

% Start (initial state 1)

        if t<=1
                k=1;
                ucom=.2;
                x1=1;
                t1=1;
        end


% Transition State Calculations

        if x2==1 & ( abs(Xd(k)-X)<=.5 & abs(Yd(k)-Y)<=.5 )
                t2=1;
        end

        if (atan2(Y-Yobs,X-Xobs)<=(psi+pi/4 | psi-pi/4)) & ((abs(X-Xobs)<=.5) & (abs(Y-Yobs)<=.
5))
                t7=1;
        end

        if x3==1 & t>= (t03+10)
                t3=1;
        end

        if  x4==1 & (abs(0-X)<=.5 & abs(0-Y)<=.5)
                t4=1;
        end

        if x5==1 & t>=t05+10
                t5=1;
        end

        if x6==1 & t>=t06+10
                t6=1;
        end

        if x2==1 & (atan2(Y-Yobs,X-Xobs)<=(psi+pi/4 | psi-pi/4)) & ((abs(X-Xobs)<=.5) & (abs(Y-
Yobs)<=.5))
                t7=1;
        end

        if x7==1 &  ucom==0
                t8=1;
        end

        if x8==1 & t>=t08+25
                t9=1;
        end

        if x4==1 & (atan2(Y-Yobs,X-Xobs)<=(psi+pi/4 | psi-pi/4)) & ((abs(X-Xobs)<=.5) & (abs(Y-
Yobs)<=.5))
                t10=1;
        end

        if x9==1 & ucom==0
                t11=1;
        end

        if x10==1 & t>=t10+25
                t12=1;
        end
```

```
                t08=0;
                psicom=atan2(Yd(k)-Y,Xd(k)-X);
                error=(psicom-psi);
                rcom=error*.3;
                ucom=.2;
        end

% Stop_And_Pickup (state 3)

        if x2==1 & t2==1
                x3=1;
                t03=t
                x2=0;
                t2=0;
                ucom=0;
        end

%  Carry_To_Drop_Off_Pile (state 4)

        if (x3==1 & t3==1) | (x10==1 & t12==1) | (x4==1)
                x4=1;
                x3=0;
                t3=0;
                t03=0;
                ucom=.2;
                psicom=atan2(0-Y,0-X)
                error=(psicom-psi);
                rcom=error*.3;
        end

%  Stop_And_Drop_Off (state 5)

        if x4==1 & t4==1
                x5=1;
                x4=0;
                t4=0;
                t05=t;
                ucom=0;
        end

%  Get_New_UXO_Position (state 6)

        if x5==1 & t5==1
                k=k+1;
                x6=1;
                t6=1;
                x5=0;
                t5=0;
                t05=0;
                t06=t;
                ucom=0;
        end



% Obstacle_Detected_Enroute_To_UXO (state 7)

        if x2==1 & t7==1
                x7=1;
                t7=0;
                x2=0;
                ucom=0;
                t08=t;
                psicom=psi+pi
                error=(psicom-psi);
                rcom=error*.3;
```

```
        end

% Obstacle_Avoidance (state 8)

        if x7==1 & t8==1 | x8==1
                x8=1;
                x7=0;
                t8=0;
                ucom=.08;
        end
        if x8==1 & t>=t08+25
                ucom=0;
                psi=psi-pi/2;
                psicom=psi;
        end

% Obstacle_Detected_Enroute_To_Disposal_Area (state 9)

        if x4==1 & t10==1
                x9=1;
                t10=0;
                x4=0;
                ucom=0;
        end

% Obstacle_Avoidance (state 10)

        if x9==1 & t11==1
                x10=1;
                t010=t;
                x9=0;
                t11=0;
                psi=psi+pi/2;
                psicom=psi;
                ucom=.08;

        end
        if x10==1 & t>=t010+25
                ucom=0;
                psi=psi-pi/2;
                psicom=psi;
        end


state=[x1;x2;x3;x4;x5;x6;x7;x8;x9;x10]

% Current System Dynamic & State Output
y=[0;rcom;ucom;0;t0;k;x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;t03;t05;t06;t08;t010]
```


STATE SPACE DISCRETE EVENT CONTROLLER
Noise (X)
Xb
Xa
-K-
Sum12
Transfer1
Noise (Y)
Yb
Ya
Sum11
Transfer
XY Location Actual
XY Location Sensed By GPS
u_slip
Sum9
Sum10
MATLAB Function
sin
cos
Product
Product1
Y(k)
Y(K+1)
Sum
Dis. Transfer
X(k)
X(k+1)
Sum1
psi(k)
psi(k+1)
Sum2
psi(k) out
u(k)
Sum3
c
.5
v1(k)
V2(k)
Dis. Transfer_
r(k)
Sum4
D
dt_
Demux
Demux1
r (turn rate)
Discrete Timer
Mux
Mux1
SLIP
intelligent controller
w1(k)
w1(k+1)
Sum6
Dis. Transfer Fcn5
W1(k)
w2(k)
w2(k+1)
Sum8
Dis. Transfer Fcn6
W2(k)
1/d1
1/d
D/d/2
D/d/2_
Sum5
Sum7
1-a1
1-a2
a
a1
rcom
ucom
k
x1
x2
x3
x4
x5
x6
x7
x8
x9
x10





VEHICLE MODELING EQUATIONS

Commanded Wheel Speed # 1 w1c(k) = ucom(k)/d + rcom(k)*D/d/2

Commanded Wheel Speed # 2 w2c(k) = ucom(k)/d - rcom(k)*D/d/2

Wheel Speed # 1 w1(k+1) = a*w1(k) + (1-a)*w1c(k)

Wheel Speed # 2 w2(k+1) = a*w2(k) + (1-a)*w2c(k)

Track Speed #1 v1(k) = w1(k)*D/d

Track Speed #2 v2(k) = w2(k)*D/d

Vehicle Speed u(k) = 0.5*(v1(k) + v2(k))

Turn Rate r(k) = 0.5*(v1-v2)

Heading psi(k+1) = psi(k) + r(k)*dt

X coordinate X(k+1) = X(k) + u(k)*(sin(psi))*dt

Y coordinate Y(k+1) = Y(k) + u(k)*(cos(psi))*dt

where,

ucom = commanded velocity (meters/sec)

rcom = commanded turn rate (rad/sec)

D = vehicle diameter (meters)

d = wheel diameter (meters)

a = motor lag constant

LIST OF REFERENCES

1. Mangolds, A., "Lemmings - A Swarming Approach to Shallow Water Mine Field Field Clearance", Report No.MCR-R0530FM-9635-693, Prepared by Foster Miller Inc., for Defence Advanced Research Projects Agency, ARPA Order No. 5916.

2. Song, S. M., Waldron, K.J., "Machines That Walk: The Adaptive Suspension Vehicle",
Advanced in Robotics, Vol. 4, No.3, pp 263-282, 1990.

3. Math Works Inc., "Student Edition of SIMULINK", Printice Hall, Englewood Cliffs, NJ., pp 3-6, 1995.

4. Ogata, Katsuhiko., "Discrete-Time Control Systems"., Printice Hall, Englewood Cliffs, NJ., pp 12-20, 1994.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road., Ste 0944
   Ft. Belvoir, VA 22304-6145

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 dyer Rd.
   Monterey, CA 93943-5101

3. Professor Anthony J. Healy, Code ME/Hy 1
   Department of Mechanical Engineering
   Monterey, CA 93943-5002

4. Department Chairman, Code ME 1
   Deparment of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5002

5. Naval Engineering Curricular Office (Code 34) 1
   Naval Postgraduate School
   Monterey CA, 93943-5002

6. Mr. Chistopher O'Donnell 3
   Naval EOD Technical Division
   Research and Development Department
   Indian Head, MD 20640

7. Mr. Arnis Manigolds 1
   Foster-Miller Inc.
   350 2nd Ave
   Waltham, MA 02143

8. Ms. Helen Griener 1
   IS Robotics
   Twin City Office Center Suite #6
   22 McGrath Hwy
   Somerville, MA 02143

9. Mr. David Kang — 1
Draper Laboratory/MS 27
555 Technology Square
Cambridge, MA 02139

10. Lieutenant Todd A. Lewis — 2
SWOS (Department Head School)
NETC
Newport, RI